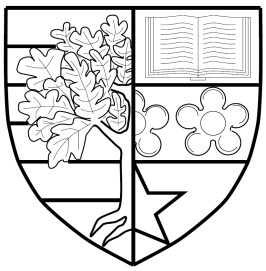
DATA-DRIVEN APPROACHES TO CONTENT SELECTION FOR DATA-TO-TEXT GENERATION

by

Dimitra Gkatzia



Submitted for the degree of
Doctor of Philosophy

Department of Computer Science
School of Mathematical and Computer Sciences
Heriot-Watt University

May 2015

To my parents, my sister and my wee nephew...

Abstract

Data-to-text systems are powerful in generating reports from data automatically and thus they simplify the presentation of complex data. Rather than presenting data using visualisation techniques, data-to-text systems use human language, which is the most common way for human-human communication. In addition, data-to-text systems can adapt their output content to users' preferences, background or interests and therefore they can be pleasant for users to interact with. Content selection is an important part of every data-to-text system, because it is the module that decides which from the available information should be conveyed to the user.

This thesis makes three important contributions. Firstly, it investigates data-driven approaches to content selection with respect to users' preferences. It develops, compares and evaluates two novel content selection methods. The first method treats content selection as a Markov Decision Process (MDP), where the content selection decisions are made sequentially, i.e. given the already chosen content, decide what to talk about next. The MDP is solved using Reinforcement Learning (RL) and is optimised with respect to a cumulative reward function. The second approach considers all content selection decisions simultaneously by taking into account data relationships and treats content selection as a multi-label classification task. The evaluation shows that the users significantly prefer the output produced by the RL framework, whereas the multi-label classification approach scores significantly higher than the RL method in automatic metrics. The results also show that the end users' preferences should be taken into account when developing Natural Language Generation (NLG) systems.

NLG systems are developed with the assistance of domain experts, however the end users are normally non-experts. Consider for instance a student feedback generation system, where the system imitates the teachers. The system will produce feedback based on the lecturers' rather than the students' preferences although students are the end users. Therefore, the second contribution of this thesis is an approach that adapts the content to "speakers" and "hearers" simultaneously. It considers initially two types of *known* stakeholders; lecturers and students. It develops a novel approach that analyses the preferences of the two groups using Principal Component Regression and uses the

derived knowledge to hand-craft a reward function that is then optimised using RL. The results show that the end users prefer the output generated by this system, rather than the output that is generated by a system that mimics the experts. Therefore, it is possible to model the middle ground of the preferences of different *known* stakeholders.

In most real world applications however, first-time users are generally *unknown*, which is a common problem for NLG and interactive systems: the system cannot adapt to user preferences without prior knowledge. This thesis contributes a novel framework for addressing *unknown* stakeholders such as first time users, using Multi-objective Optimisation to minimise regret for multiple possible user types. In this framework, the content preferences of potential users are modelled as objective functions, which are simultaneously optimised using Multi-objective Optimisation. This approach outperforms two meaningful baselines and minimises regret for *unknown* users.

Acknowledgements

First of all, I want to express my gratitude to my supervisor, Helen Hastie, for introducing me to the field of Natural Language Generation, for her guidance and assistance throughout my PhD and for providing invaluable advice. I would also like to thank my second supervisor Oliver Lemon, for always being there to offer an extra point of view, a new insight and new ideas. I would also like to thank Verena Rieser for providing me with constructive feedback during my PhD and for giving me the opportunity to work on the ATME project, and later on the GUI project. I am very grateful to Alasdair Mort for sharing the MIME dataset with us and to Sandy, Alistair and Micha for all the fruitful conversations. I am very grateful to my thesis examiners, Fiona McNeill and Yaji Sripada, for providing constructive feedback and meaningful comments which helped shaping this thesis.

During my PhD, I was blissful enough to work within the Interaction Lab, at Heriot-Watt University. I had the chance to meet brilliant researchers and learn a lot from the *interaction* with them. Many thanks to my HW colleagues: Srini, Patricia, Mary Ellen, Marta, Mathias, Eshrag, Lilia, Tom and Zhuoran for collaborating on various research, teaching and organisational tasks and for discussing several research issues. I would like to thank David McGookin for inviting me to Aalto University, where I had the chance to work on an exciting project and broaden my research interests. Finally, I am very grateful to Simon Keizer, Effie Bellou and Nikos Aletras for reading parts of my thesis and providing useful comments.

I couldn't survive the last three and a half years without the support of my close friends and family. Anastasia, Effie and Xara, thank you for the encouragement, for the support and for managing to share great moments even if we were living in different parts of the world ♣. I would also like to thank Stefano for all the support, encouragement and useful advice. Last, but not least, I would like to thank my parents and my sister for always supporting my educational endeavours and for always being optimistic and encouraging.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Data-to-text generation is the subfield of Natural Language Generation (NLG) that addresses the task of automatically generating text from data, such as sensor data or event logs (Reiter, 2007). Data-to-text systems consist of several distinct modules such as data analysis, content selection (what to say) and surface realisation (how to say it) (Reiter, 2007). This thesis focuses on the task of content selection from time-series data.

Time-series data such as sensor data, weather data and stock market data, often display a complex structure and the identification of useful information is domain-dependent. Humans who are experts in their domain can describe time-series data fluently by using natural language. Their descriptions are a result of a combination of decisions, for instance, they can decide to refer to the unusual fluctuations, the averages or the time-series changes (trends) over time. Moreover, they can decide to refer to the data in a sensible order, or to use their broad knowledge to justify and/or explain the time-series data. However, for machines, the task of determining the content to effectively summarise time-series data in natural language remains challenging, due to the fact that content determination in general is domain-dependent. The necessity of general content selection models has been acknowledged by the NLG community (Bouayad-Agha et al., 2012).

The motivation of developing data-to-text systems and focusing on the task of con-

tent selection is described in the next section (Section 1.1). Section 1.2 presents the challenges of data-driven adaptive NLG systems and how they are addressed in this thesis. Section 1.3 refers to the research questions that are explored in this thesis. Sections 1.4 and 1.5 refer to the contributions made to the field and the publications resulted from this research respectively. Section 1.6 provides a thesis overview.

1.1 Motivation

This section motivates the work of the thesis. In particular, it aims to answer: (1) why content selection is useful for data-to-text systems (Section 1.1.1); (2) why it is preferable to present textual summaries of time-series data rather than graphical representations (Section 1.1.2); and (3) why it is important to adapt the selected content to different user groups (Section 1.1.3).

1.1.1 The Importance of Content Selection

Natural Language Generation can be used for simplifying the presentation of complex data. The overload of time-series data available through the web, sensors and other means has increased the need for humans to digest these time-series data in an automatic, accurate and time-efficient manner. NLG systems can deal with this challenge in an automatic way. For example, an NLG system can read sensor data and produce a comprehensive textual summary. Content selection is an important part of every NLG system, because it is the module that decides which from the available information should be conveyed to the user. Previous research on generation from time-series data has been conducted in several domains such as weather forecasts (Sripada et al., 2004; Konstas and Lapata, 2012), health informatics (Gatt et al., 2009), stock market summaries (Kukich, 1983) and assistive technology systems (Black et al., 2010), as we will discuss in further detail in Chapter 2. These systems have employed different content selection methods, which are reviewed in Chapter 2. In this thesis, we present a com-

parison of two promising data-driven methods for content selection and we present the results in two experimental domains: student feedback and health informatics (Chapter 4). We further argue that it is essential for data-to-text systems to choose the relevant information to communicate so as to satisfy the conflicting preferences of stakeholders, such as lecturers and students. Although NLG systems have the potential to assist in decision making (Gatt et al., 2009), this thesis focuses on users' preferences. We further demonstrate a method that addresses preferences of unknown users (Chapter 6).

1.1.2 Effectiveness of Textual Summaries over Graphical Representations

Data and in particular time-series data are normally presented using visualisation techniques that can be difficult for an inexperienced user to understand and interpret. Natural Language Generation faces the challenge of communicating the data in a simpler, more effective and more understandable way, by conveying information through language. Recent studies have demonstrated that text descriptions can be more effective, understandable and helpful in decision making than graphical representations of sensor data (Law et al., 2005; van den Meulen et al., 2010).

Early research has shown that graphs require expertise in order to be interpreted (Petre, 1995). There are three recent studies that compare graphs to textual summaries in terms of effectiveness in decision making in the clinical domain, as described below:

1. Law et al. (2005) show that clinical staff tend to make more correct clinical decisions when viewing textual formats of data rather than when consulting graphs.

2. In a similar study, van den Meulen et al. (2010) show that users prefer the textual descriptions produced by humans more than the computer generated textual descriptions. In addition, they show that, in decision making, the computerised textual reports are as useful as the graphical representations that the clinical staff were familiar with.

3. Gatt et al. (2009) show that all users (doctors and nurses) perform better in parallel tasks (making decisions after viewing a text summary vs. a graphical representation) with human-written texts rather than graphs. Compared to computer-generated summaries, they perform worse than they perform with the hand-written texts. The users find the computer-based texts as useful as the graphs.

These studies show that the interpretation of graphical representations is not always obvious. It is also shown that textual descriptions can support decision making more effectively than graphical representations. In addition, during our data collection (Chapter 3), subjects report that it is difficult to link information from nine graphs simultaneously. Finally, for tasks such as automatic student feedback production, the textual summaries should be enhanced with relevant statements, such as motivational phrases or advice, which is infeasible to be depicted on graphs. Therefore, textual summaries can improve decision support systems and enhance the understandability of time-series data. Although we acknowledge the importance of combining text and visualisations (e.g. (Mahamood et al., 2014; Sripada and Gao, 2007)), in this thesis we focus only on automatic content selection for NLG from time-series data.

1.1.3 User-adaptive Output

Different user groups such as doctors/nurses/parents (in medical domain) or lecturers/students (in student feedback) or expert/non-expert users (in various domains) have different information needs and preferences, therefore personalised reports are important. Hunter et al. (2011) emphasise that personalisation should be based on relevant factors/variables and not only demographic data[1]. Users have different preferences and goals and the systems should adapt to those in order to be preferred. DiMarco et al. (2008) emphasise that it is necessary for a system to avoid referring to events that seem irrelevant for the majority of the users, but are relevant for a particular user.

[1]In Chapter 3, we also show that users' preferences are independent of gender, occupation and previous experience.

For example, today's health care systems may provide too much irrelevant information to patients or omit important information, which leads the users to believe that the system is not addressed to them (DiMarco et al., 2008). This can have negative impact on the patients' compliance with medical regimens etc. Similarly, in the student feedback domain, a general system would advise students to study x hours per day. For a hard-working student this advice might be irrelevant or even confusing.

On a related note, our experiments (Chapter 4) show that there is a mismatch between the preferences of students and lecturers on what constitutes a good feedback summary. We tackle this challenge by introducing a new task: Multi-adaptive Natural Language Generation (MaNLG - Chapter 5), which aims to find middle ground between the preferences of "speakers" and "hearers", as for instance, lecturers and students, or patients and doctors. We further show that this methodology can be adapted for dealing with unknown users (Chapter 6).

1.2 Challenges for Data-driven Adaptive NLG Systems

In this section, we present the challenges for developing NLG systems and how we deal with them in this thesis.

- Data availability: The lack of aligned datasets (data and corresponding summaries) which can be used to derive rules or to train an NLG system is a major challenge for NLG engineers. Although data is widely available, they cannot be used directly for the development of an NLG system, because there is lack of alignment between input and output data, as for instance expert written summaries (Belz and Kow, 2010). Data-driven data-to-text systems require large corpora with data that can be aligned to natural language text so as to be used as an input to a training algorithm. To overcome this barrier in our domains, we conducted two data collections in order to acquire the relevant information (Chapter

3).

- Lack of or inconsistent expert knowledge: Another issue is the lack of expert knowledge or the difficulties of acquiring it due to several factors, such as difficulties in recruiting experts. The main challenge that we face in this work is that experts provide a variety of responses. This process introduces difficulties in knowledge acquisition. This challenge has been also noted by Sripada et al. (2004). We handle this challenge in three different ways: (1) by applying multi-label classification which is able to handle mis-matches in aligned corpora (Chapter 4); (2) by asking users to rate expert constructed and random summaries in order to derive their preferences, similar to Rambow et al. (2001) (Chapters 3, 4 and 5); and (3) by clustering the experts' responses so as experts with same preferences belong to the same cluster (Chapter 6).

- Evaluation challenges: As other areas of Computational Linguistics, NLG also suffers from the limitations of the available evaluation methods. Reiter and Sripada (2002) firstly questioned the suitability of corpus-based approaches to evaluation of NLG systems, followed by Belz and Reiter (2006) and Foster and Oberlander (2006). Text corpora from data are usually gathered by asking experts to provide written textual summaries or descriptions. However, experts use different words to communicate the data or they choose to refer to different events, which makes it difficult to construct a consistent dataset and therefore using it as gold standard for evaluation. To tackle this issue, we perform user studies in all of our experiments and in some cases, where it is sensible, we use automatic evaluation as well (Sections 4.1.5, 4.3.2, 5.3, 6.3).

- Lack of prior knowledge about the users: One of the most crucial issues in user-adaptation is the lack of prior knowledge of the users. Previous approaches to tackling this issue include the use of latent User Models (Han et al., 2014) initial questionnaires to derive information by the user (Reiter et al., 1999) and dynamic

user modelling (Janarthanam, 2011). These approaches require prior interaction between the user and the system. In this thesis, we tackle this problem by asking potential users to rate several summaries with different content and thus derive their preferences. The potential users are then grouped in terms of preferences and we use Multi-objective optimisation to simultaneously adapt to all preferences (Chapter 6).

1.3 Research Questions

This section states the research questions explored in this thesis.

- RQ 1: With respect to user preferences, can the task of content selection be formulated and solved effectively using different data-driven techniques and hand crafted methods (Chapter 4)?
- RQ 2: Can we simultaneously adapt content to different *known* stakeholders (Chapter 5)?
- RQ 3: Can we effectively address *unknown* users or stakeholders, i.e. users with unknown preferences or group membership (Chapter 6)?

1.4 Contributions

This thesis seeks to develop novel approaches to address the research questions presented in the previous section. Its contributions are as follows:

1.4.1 Comparison of Data-driven Approaches to Content Selection for Data-to-Text Systems - RQ 1

The task of content selection can be treated as a data-driven task in different ways. Since supervised learning and Reinforcement Learning have different theoretical and

practical foundations and have been both used for NLG (see Chapter 2), it is of interest to directly compare them in a data-to-text system. *Multi-label classification* is able to capture dependencies between the data and the content by making the content selection decisions simultaneously. In addition, due to its ability to split the classification task into subtasks, it can easily work well with limited data. *Reinforcement Learning* treats the summarisation of time-series data as a sequential problem, where each decision is affected by the previous. Our user evaluation (Section 4.3) shows that multi-label classification makes decisions similarly to the ones observed in the expert written dataset, whereas Reinforcement Learning is able to produce more variable output.

1.4.2 Multi-adaptive Natural Language Generation - RQ 2

In this thesis, we present a novel challenge, *Multi-adaptive Natural Language Generation (MaNLG)*, which refers to the task of automatically adapting the content to different stakeholders such as "speakers" and "hearers" or lecturers and students. Existing methods for adaptive NLG use User Models (UMs) or different versions of a system in order to deal with different users. However, these methods assume prior knowledge of the users, which is not always applicable in real world applications, such as health monitoring systems or e-learning applications. We show that we are able to find middle ground between the preferences of students and lecturers, and thus generate textual output that is preferable by both groups. The task of MaNLG is treated as a combinatorial problem, where the most important preferences to be optimised are identified through Principal Component Regression. We evaluate our methodology in the domain of student feedback generation. In the experimental setup, lecturers and students are asked to rank different summaries based on the same data. The summaries are generated by the following systems: (1) student-adapted system; (2) lecturer-adapted system; and (3) a PCR-based system. It is shown that both groups mostly prefer the system that adapts to their group. Lecturers however rate the PCR-based system similarly to the Lecturer-adapted system, which indicates that the PCR-based system is of high quality

even if it does not adapt to lecturers' preferences. Students, who are the end-users of the feedback generation system, rated the PCR-based system significantly higher that the Lecturer-adapted system, which indicates that a system can be improved by taking into account the preferences of the end users.

However, we can not always distinguish users in terms of group membership or occupation. In the following section, we describe how we address unknown users with a multi-adaptive approach.

1.4.3 Addressing Unknown Users - RQ 3

Previously, the stakeholders could be distinguished between lecturers and students. However, in most real world scenarios, users are generally unknown without prior interaction with the system. Therefore, we present a method that is able to address unknown users. Particularly, we perform an experiment in the health informatics domain, where we summarise physiological sensor data from people who require first aid. The idea behind the overall project is to develop a system that provides decision support in a first aid scenario. The system should address users with different levels of expertise, from medical doctors to users without prior experience in first aid. In this scenario, time is critical and user profiling cannot be performed at the time of a casualty, i.e. it would be inappropriate to ask users about their background during a medical emergency. As such, our system optimises the output with respect to a pool of potential users. As potential users have different preferences regardless of their profession, prior experience with sensor data or gender, we cluster the users regarding their preferences and then we apply Multi-objective Optimisation (MOO) to simultaneously adapt to various groups (Chapter 6).

1.5 Publications

Part of the work presented here has been published and presented in peer-reviewed conferences and workshops:

1. Dimitra Gkatzia, Verena Rieser, Alexander McSporran, Alistair McGowan, Alasdair Mort and Michaela Dewar. Generating Verbal Descriptions from Medical Sensor Data: A Corpus Study on User Preferences. In Proceedings of BCS Health Informatics Scotland (HIS). Glasgow, UK, 2014 (Gkatzia et al., 2014d). (Chapters 3 and 6).

2. Dimitra Gkatzia, Helen Hastie and Oliver Lemon. Comparing Multi-label classification with Reinforcement Learning for Summarisation of Time-series data. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL). Baltimore, USA, 2014 (Gkatzia et al., 2014a). (Chapter 4).

3. Dimitra Gkatzia, Helen Hastie and Oliver Lemon. Multi-adaptive Natural Language Generation using Principal Component Regression. In Proceedings of the 8th International Natural Language Generation Conference (INLG). Philadelphia, USA, 2014 (Gkatzia et al., 2014c). (Chapter 5).

4. Dimitra Gkatzia, Helen Hastie and Oliver Lemon. Finding Middle Ground? Multi-objective Natural Language Generation from time-series data. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL). Gothenburg, Sweden, 2014 (Gkatzia et al., 2014b). (Chapter 5).

5. Dimitra Gkatzia, Helen Hastie, Srinivasan Janarthatanam and Oliver Lemon. Generating student feedback from time-series data using Reinforcement Learning. In Proceedings of the 14th European Workshop on Natural Language Generation (ENLG). Sofia, Bulgaria, 2013 (Gkatzia et al., 2013). (Chapter 4).

6. Dimitra Gkatzia and Helen Hastie. Dynamic user modelling for personalized report generation of time-series data. In Symposium on Influencing People with Information (SIPI). Aberdeen, Scotland, 2012 (Gkatzia and Hastie, 2012). (Chapter 5).

Non refereed

1. Dimitra Gkatzia and Helen Hastie. An Ensemble Method for Content Selection for Data-to-text Generation. In 1st International Workshop on Data-to-text Generation, Edinburgh, UK, 2015 (Gkatzia and Hastie, 2015). (Chapter 4).

2. Dimitra Gkatzia. "Keep up the good work!" Generating Feedback for Students using Reinforcement Learning. In SICSA PhD Conference. Stirling, UK, 2013 (Gkatzia, 2013). (Chapter 4).

1.6 Thesis Overview

This thesis makes a contribution to the field of content selection for adaptive and non-adaptive Natural Language Generation systems. The remainder of the thesis is organised as follows and is depicted in Figure 1.1:

- **Chapter 2** discusses the background; essential terminology of Natural Language Generation and previous approaches to content selection. The approaches are broken down to rule-based and data-driven approaches. A review of user-adaptive approaches to NLG is given. Evaluation methods for NLG systems are also discussed. Finally, the chapter concludes with a critical analysis of current practices.

- **Chapter 3** describes the overall framework developed for the experiments of this thesis. In addition, emphasis is given to two data collections performed in the domains of (1) student feedback generation and (2) health informatics, followed by corpus analysis and discussion.

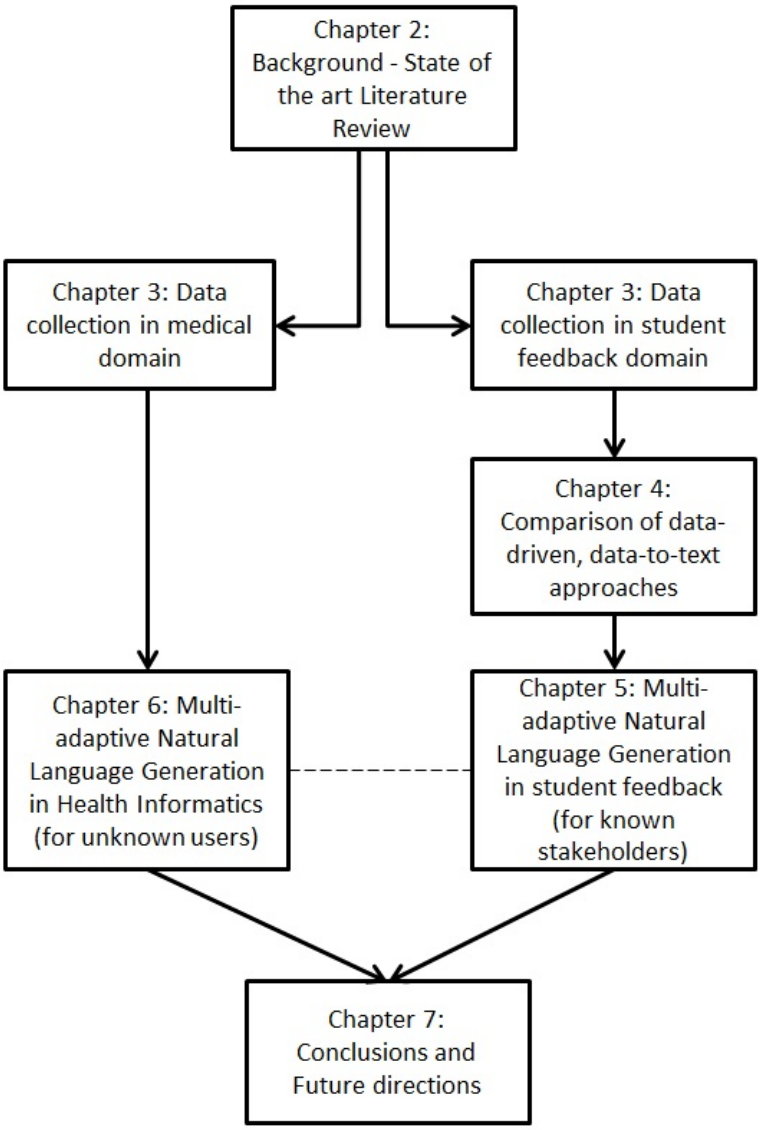


Figure 1.1: Thesis structure.

- **Chapter 4** develops and compares novel approaches to content selection from time-series data: (1) a rule-based approach; (2) a Multi-label Classification approach (supervised learning); and (3) a Reinforcement Learning approach (Temporal Difference learning). The benefits and limitations of each approach are elaborated.

- **Chapter 5** introduces a new challenge, *Multi-adaptive Natural Language Generation (MaNLG)*, describes the motivation and develops an approach for tackling this task.

- **Chapter 6** develops an approach for addressing unknown users using multi-objective optimisation in the health informatics domain.
- **Chapter 7** summarises the main findings and contributions of this thesis and suggests possible avenues for future work.

Chapter 2

Literature Review

This chapter describes the related work on content selection in the context of data-to-text systems, which draws onto the motivation and the challenges of developing data-to-text systems as discussed in the introduction. It classifies the related work into two main categories: rule-based approaches and data-driven methods. It presents an overview of the state-of-the-art methods and critically analyses the strengths and limitations of each category. Since user-adaptation is a matter that affects not only NLG systems but also other interactive systems, this chapter also discusses user-adaptive NLG and interactive systems.

Specifically, the chapter begins by presenting the current state-of-art data-to-text system architecture and introducing the appropriate terminology in Section 2.1. Section 2.2 describes rule-based approaches to content selection and Section 2.3 refers to data-driven approaches to content selection. Next, Section 2.4 discusses adaptive NLG systems, followed by a discussion on evaluation metrics (Section 2.5). Finally, Section 2.6 concludes the chapter by critically discussing the different approaches.

2.1 Data-to-text System Architecture

A fundamental decision in data-to-text system development concerns the system architecture. Data-to-text systems need to perform multiple tasks such as data analysis,

content selection etc. The mainstream data-to-text architecture is a pipeline architecture which is proposed by Reiter (2007). It consists of four distinct modules: (1) Signal Analysis, (2) Data Interpretation, (3) Document Planning and (4) Microplanning and Realisation. Briefly, the four modules are described below and their relations are depicted in Figure 2.1:

1. **Signal Analysis:** The Signal Analysis module is responsible for analysing the input data, identifying patterns and trends. This is an essential part of a data-to-text system when the input is numerical data.

2. **Data Interpretation:** The Data Interpretation module is responsible for detecting causal and other relations between the patterns and trends identified by the Signal Analysis module. This module is useful for NLG systems that aim to communicate more complex messages, such as explanations.

3. **Document Planning:** The Document Planner decides which of the identified patterns, trends and relations should be mentioned in the generated text, a task known as *content selection*. It is also responsible for structuring the generated text, i.e. deciding on ordering the information, the paragraph breaks in longer generated documents and the general structure of a document. The document planner is essential when part of the available content needs to be communicated, such as in report generation or summarisation systems.

4. **Microplanning and Realisation:** This module actually generates the output text. Every NLG system contains a realisation module.

This thesis adapts the general architecture to the needs of our domains, as discussed in Chapter 3 - page 39. The focus of the research described in this thesis is on the task of content selection from time-series data. As the targeted generated texts are a paragraph long, there is no need for deciding on document structuring such as paragraph breaks. The rest of the components are described in Chapter 3.

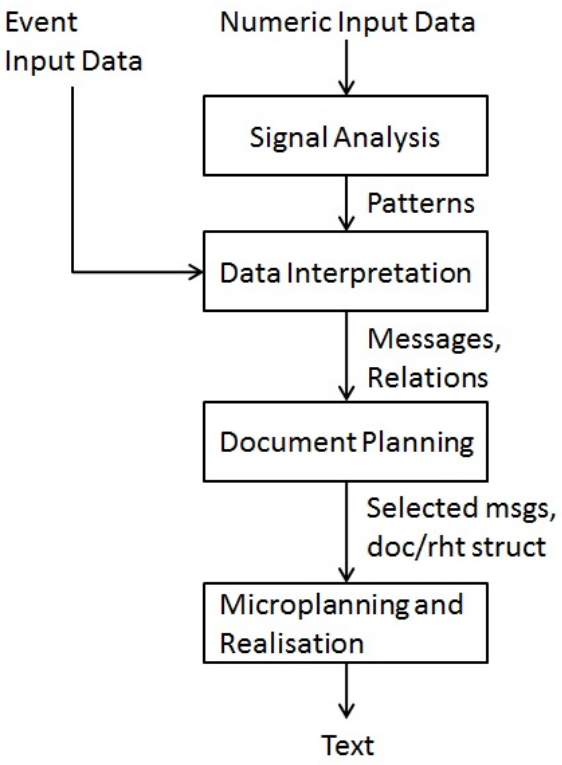


Figure 2.1: Data-to-text system architecture Reiter (2007).

2.2 Rule-based Content Selection in Data-to-Text Systems

This section describes previous work that treats content selection in a rule-based manner. Rule-based systems generate understandable output, are thoroughly studied, are robust in small domains and are suitable for commercial use. Table 2.1 summarises the methods and application domains of these systems. One of the earliest data-to-text applications is TREND (Boyd, 1998). TREND includes a detailed module for time-series analysis using wavelets. However, this system does not include a notion of content selection, as it is mostly focused on describing all trends that are observed in data.

Sripada et al. (2001) suggests a two-stage model for content selection from time series data (sensor readings from a gas turbine and numerical weather simulations). The model assumes that the data source is an external component and that a *Domain Reasoner (DR)* module is present. The DR is responsible for making inferences. The

Author(s)	**Method**	**Domain**	**Data Source**
Boyd (1998)	No content selection	Weather	database
Sripada et al. (2001)	Two stage model: (1) Domain Reasoner and (2) Communication Reasoner	Weather, Oil rigs	sensors, numerical data
Sripada et al. (2003)	Gricean Maxims	Weather, Gas turbines, Health	sensors
Hallett et al. (2006)	Rule-based	Health	database
Yu et al. (2007)	Rules derived from corpus analysis and domain knowledge	Gas Turbines	sensors
Sripada and Gao (2007)	Decompression Models	Diving	sensors
Gatt et al. (2009)	Rule-based	Health	sensors
Thomas et al. (2010)	Document Schemas	Georeferenced Data	database
Peddington and Tintarev (2011)	Threshold-based rules	Assistive Technology	sensors
Demir et al. (2011)	Rule-based	Domain independent	graphs - database
Johnson and Lane (2011)	Search Algorithms	Autonomous Underwater Vehicle	sensors
Banaee et al. (2013)	Rule-based	Health	grid of sensors
Schneider et al. (2013)	Rule-based	Health	sensors

Table 2.1: Content selection in rule-based data-to-text systems

inferences together with the system's communicative goal are used for building an overview of the summary. Finally, the *Communication Reasoner* module takes as input the output of the DR and it specifies the final content, which is then available to the other NLG tasks, i.e. microplanning and surface realisation. It is worth mentioning that this approach was suggested before Reiter's (2007) architecture (page 14).

Sripada et al. (2003) introduces a Gricean Maxims-based approach to Natural Language Generation from the same data sources as previously mentioned plus medical sensor data. The Gricean maxims are used in order to communicate the content, after the segmentation algorithms have been applied for data analysis (Sripada et al., 2003).

The maxims reflect the cooperative principle that describes how people communicate and act with one another, by using utterances, their flow and their meaning. The Gricean maxims (Grice, 1975) constitute the Quality, Quantity, Relation and Manner maxims and they are inspired by the pragmatics of natural language. The maxim of Quality influences the content selection decisions regarding the real values of the data by using linear interpolation. The maxim of Quantity decides on which patterns and trends are useful for the user. The maxim of Relevance states that the information should be relevant to a particular user and User Models are acquired for this task, as in Reiter et al. (2003). Finally, the maxim of Manner states that the information should be conveyed in the most appropriate linguistic manner, without ambiguity, briefly and in a sensible order.

Hallett et al. (2006) presents a content selection approach for summarisation of medical histories which is based on two elements: (1) the type of the summary and (2) the length of the summary. They also introduce a list of concepts and events that are linked to those concepts. In the content selection phase the events are clustered in terms of relevance. It is assumed that smaller clusters do not include important events and, therefore, only the larger clusters of events are mentioned in the summary. Depending on the type and the length of the summary the content attributes are determined in a rule-based fashion. For instance, a medical condition might be a main event and its attributes could be name, status, clinical course etc.

Sripada and Gao (2007) report the ScubaText system which generates reports from scuba-dive computer data and improves the safety of the dives. The data analysis module determines the interpretations of the patterns identified regarding the safety of dives. Decompression models (similar to those used by dive computers) are used to generate recommendations on when the bottom of the sea is safe for diving. Using these interpretations, deviations from the actual dive are computed. Then ratings are assigned inversely proportional to the deviations and they influence the text generation decisions.

Yu et al. (2007) describe SumTime-Turbine, a system that generates summaries of large time-series data sets from gas turbines' sensors. This system adopts a bottom-up approach, where the NLG system emerges by joining subsystems together. It consists of two main components: a data analysis module that is responsible for content determination and the Natural Language Generation module. The data analysis component can be further split up into:

- Pattern Recognition, which is responsible for connecting time-series segments to concepts.
- Pattern Abstraction, which produces high level abstractions of the patterns.
- Interesting Pattern Selection, which is responsible for deciding which of the abstract patterns should be conveyed in the summary. The content is determined by using domain knowledge and historical pattern frequency.

The content order is based on rules obtained via corpus analysis and is inspired by the way that experts tend to summarise sensor data. In particular, the content follows the following ordering.

1. Background information
2. Overall Description
3. Most significant patterns

In this thesis, content ordering is determined in a similar fashion. In the domain of student feedback generation, the order is defined by observed pattern history and in health informatics it is determined by consulting experts (domain knowledge).

The BabyTalk (BT) system (Gatt et al., 2009) produces textual summaries of data in the context of a neonatal intensive care unit. The data used as input consists of (1) sensor data (Heart Rate, mean Blood Pressure and Oxygen Saturation), (2) lab results and observations, (3) events such as nurses' actions, medical diagnosis and treatment

and other information, and (4) free text. The BT-45 system (Portet et al., 2007) generates a summary after 45 minutes of measurements and the collection of the data mentioned earlier. Its aim is to interpret the data by linking events to observations, not to offer diagnosis. Content selection is handled as described in Hallett et al. (2006). The length of the goal summary is a deciding factor as well as the type of problem. The events are assigned a value which represents their importance in the data interpretation module, but this value is not updated with respect to the selected content, e.g. an event that explains a fact may be omitted because the data interpretation module initially assigned a low value and because this value is not updated, the output summary might not be coherent. The events are clustered in terms of relevance and the first step of the summarisation dictates the removal of the smaller clusters, because they are usually irrelevant. Next, the important events and the level of detail are influenced by the relevance to the type of summary. The system's drawback is that it does not update the importance of the events (regarding the probability to be selected) after one event is being selected.

Black et al. (2010) develop a story generation rule-based system that is addressed to children with complex communication needs. The input of this system is non-linguistic data gathered through sensors which describe the child's location, activities and interactions with people or objects. Specifically, the data are collected through: (1) RFID readers that monitor the places that the child visits, (2) a microphone that is used for recording events, and (3) a visual interface with an access switch that the child can use with its head. The teacher and the school staff can also enter information about the child's activities. Figure 2.2 depicts the overall system.

The aim of the system is to generate a story that describes "how was school today...". The system groups elements into events in order to determine the content (Peddington and Tintarev, 2011), by using clustering algorithms that classify events depending on the location, the time and the voice recordings. It also uses rules to define unexpected events, for example the divergence with the child's usual timetable and activities. The

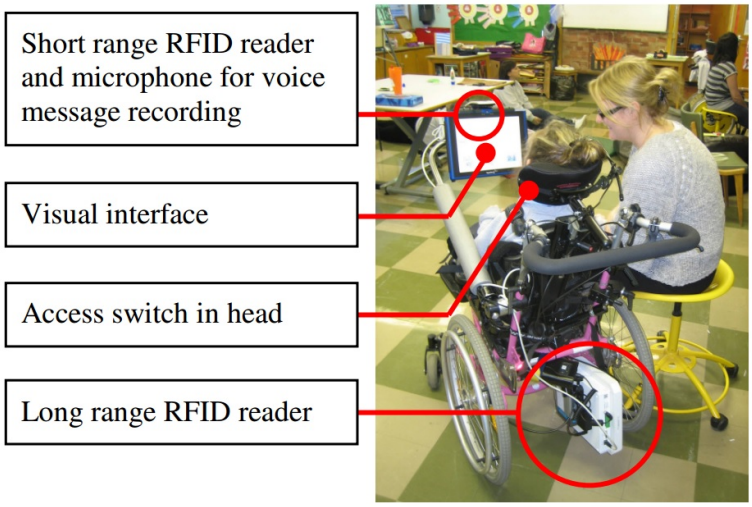


Figure 2.2: The "How was the school today...?" system Black et al. (2010).

derived rules are based on a User Model that takes into account the cognitive model, the timetable, unexpected events and inherent "interestingness".

Demir et al. (2011) presents an approach to summarisation of bar charts. This is a domain-independent approach which is based on users' scores of potential content to be present in a brief summary. After averaging scores from a data collection, rules were derived that determine what information in a graph should be included in the summary. Similar to this method, we collected user ratings in order to be able to adapt to user preferences.

Johnson and Lane (2011) present Glaykos, a system that automatically generates audio visual debriefs for underwater missions. The data used are collected through an Autonomous Underwater Vehicle (AUV) that is armed with sensors. The sensor data describe attributes of the bottom of the sea. In order to generate the multimodal output, a situation model is used, which consists of a bitmap situation model and a vector situation model in order to include all the data from the underwater mission and the related simple and complex concepts. Initially, the data from the mission are processed and linked with the other elements of this bitmap situation model. From this, a vector situation model is created, which models the motivation and the causation models. Next, the events are ordered and grouped together according to the time

they happened (adjacent time), whether they have the same actor, whether they do not have contradictory motivations and whether the first event causes the following event and not anything else. These groups are viewed as an instance of the travelling salesman problem, where each group represents one city. In order to solve this, two optimisation algorithms have been applied, a depth-first search and a genetic algorithm. Both algorithms used the same fitness function, which is based on the spatiality of the event (a penalty is given if it is in a different region), temporality of the action, the protagonist, the motivation and the causality. In Chapter 6, genetic algorithms are also used as part of a multi-objective framework for NLG.

Banaee et al. (2013) presents a content selection approach for summarisation of physiological sensor data based on the importance of potential content messages: (1) global information, event based, and (3) summary based messages. Each message category uses a ranking function to assign an "importance" value to the message. The ordering of the message is based on how important the message is and whether there are dependencies between messages.

Schneider et al. (2013) describe an approach to summarising medical sensor data in pre-hospital care (MIME project). The content selection module is rule-based and it uses trees that associate the chosen information, based on the Rhetorical Structure Theory (Mann and Thompson, 1988). The rules are derived through a combination of corpus analysis and expert consultation. A subset of the scenarios used in the MIME project have been used in Chapter 6. However, the generation goals differ, therefore a direct comparison is not applicable. Deriving rules by working with experts is a common practice in developing rule-based data-to-text systems. In Chapter 4, we describe a baseline rule-based system, which is developed by acquiring knowledge from an expert. We further incorporated end user's potential preferences by asking a student to provide his preferences.

This section presented several rule-based approaches to content selection for data-to-text systems. As these approaches have been used for different systems in various

domains, comparisons are not applicable, because rules are not transferable between systems and domains and the systems are not available for research or other purposes. However, there are good practices and lessons learnt from this previous work:

1. NLG rule-based systems are thoroughly studied and developed by taking the end-user into account.

2. NLG is part of a wider software system and it is difficult to be evaluated solely.

3. Rule-based system are robust and therefore they have been used for commercial reasons as for instance `www.metoffice.gov.uk/invent/data2text`.

The next section discusses data-driven approaches to NLG and a comparison will be thoroughly given in Section 2.6.

2.3 Data-driven/ Trainable Approaches to Content Selection

This section discusses data-driven/trainable approaches to content selection. A subset of data-driven approaches to Natural Language Generation treats content selection and surface realisation in a unified manner. Therefore, here we discuss systems that *learn* how to choose content, either as an independent task or jointly with surface realisation. Table 2.2 summarises data-driven approaches to NLG. Trainable NLG has been mostly applied in interactive settings, therefore, in the next section we will refer to the state-of-art real-time generation.

Data-driven approaches to NLG have been initially introduced at the sentence level (Knight and Hatzivassiloglou, 1995). Langkilde and Knight (1998) introduce a trainable approach to Natural Language Generation, which works in two steps. Firstly, possible utterances are generated and secondly, they are ranked according to probabilities derived through corpus analysis. Mellish et al. (1998) describe a similar stochastic approach based on generate-and-rank. The technique is applied in the context of text

Author(s)	Method	Task	Domain
Mellish et al. (1998)	Generate-and-rank	Content Selection	Item descriptions
Duboue and McKeown (2002)	Genetic Algorithms	Content Selection	Health
Duboue and McKeown (2003)	Classification	Content Selection	Biographical Descriptions
Turner et al. (2008)	Decision Trees	Georeferenced Data	database
Barzilay and Lapata (2005)	Classification	Content Selection	Sports
Barzilay and Lee (2004)	Hidden Markov Models (HMMs)	Content Selection, Ordering, Summarisation	Earthquakes, Clashes, Drugs, Finance, Accidents
Liang et al. (2009)	HMMs	Content Selection	Sportscasting, Weather
Angeli et al. (2010)	HMMs with Log-linear models	Content Selection	Sportscasting, Weather
Konstas and Lapata (2012)	Structured Perceptron	Content Selection	Flights
Lampouras and Androutsopoulos (2013)	Integer Linear Programming	Content Selection, Lexicalisation and Sentence aggregation	Wine descriptions
Kondadadi et al. (2013)	Support Vector Machines	Content Selection, Realisation	Biography, Weather
Sowdaboina et al. (2014)	Neural Networks	Content Selection	Weather

Table 2.2: Data-driven approaches to content selection in data-to-text systems. In all cases, the data sources are database entries, apart from Lampouras and Androutsopoulos (2013) who use ontologies.

planning and is used to select the best of the candidate solutions (candidate solutions are generated and then the one to be present in the output is selected stochastically). A similar approach has been later used by Stent et al. (2004) for sentence level generation. Similarly, Duboue and McKeown (2002) present an approach to content planning using Genetic Algorithms that is able to identify common patterns in the data.

Duboue and McKeown (2003) present a content selection approach where the available content consists of a corpus of text expressed as semantic features. They treat

content selection as a classification task where the objective is to decide whether a database entry should be included in the output or not. In this thesis, one of our baseline systems follows similar structure. As it is not possible to reuse this system, we also formulated the task of content selection in the same way in order to emulate this system. Turner et al. (2008) present a decision tree approach to content selection in the domain of description generation of georeferenced data. In this framework, content is represented as leaves of a tree, whereas the nodes represent events. The text is then generated from the content that exists in leaves. In Chapter 4, decision trees are used as a baseline system (Section 4.2.2). In a similar domain, Thomas et al. (2010) use document schemas to induce document plans for textual descriptions of georeferenced data for blind users. The selection of the schema is influenced by the spatial data analysis. The drawback of such systems is that content is selected separately without accounting for data dependencies.

Barzilay and Lapata (2005) overcome this limitation by developing a collective content selection approach which is a classification task that makes content decisions in a collective way. Their approach initially considers an individual preference score, which is defined as the preference of the entity to be selected or omitted and it is based on: (1) the values of entity attributes and is computed using a boosting algorithm and (2) the identification of links between the entities with similar labels. This method has been applied in sports domain where the data can be related in a timely manner, i.e. one player's action can cause the injury of another. The collective content selection approach differs from Duboue and McKeown (2003) approach in that it allows contextual dependencies because the entries are selected depending on each other and not in isolation. In Chapter 4, we present a multi-label classification approach that is based on the same vein, i.e. the content selection decisions are not made independently. Unfortunately, the dataset used by Barzilay and Lapata (2005) is not publicly available and therefore a direct comparison is impossible. We believe that, because multi-label classification and collective content selection have a similar theoretical background,

comparing them with the same dataset would be of great interest.

Barzilay and Lee (2004) treat content selection as Hidden Markov Models, where states correspond to information types and state transitions define the potential ordering. The state transition probabilities define the chance to change from a given topic to another. Liang et al. (2009) present a model for generation using a 3-tier HMMs in order to address the task of segmenting the utterances, mapping the sentences to meaning representations and choosing the content for generation. The aim of this model is to effectively cope with the segmentation, the grouping of relevant facts, and the alignment of the segmentation results to facts. For this purpose, it is assumed that a world state is represented by records and text, and each record is comprised of fields and their values. For example, in the weather domain, the text is the weather forecast, the records are the different weather attributes such as rain chance, temperature or wind speed, the fields can be the maximum or minimum temperature or wind speed and the values the numerical or categorical values. The parameters of this model are calculated through an Expectation Maximisation algorithm and the model is tested in three domains in order to prove its generic nature: Robocup Sportcasting, Weather Reports and NFL Recaps. The process starts with the record selection (e.g. the temperature is selected for generation), then the field selection (e.g. the minimum temperature), and finally the word selection to be generated (e.g. the numerical values of the minimum temperature). The drawback of this model is that it does not treat record, field and word choices in a unified manner so as to capture potential dependencies. Therefore, Angeli et al. (2010) extend this model in order to capture the dependencies between records, fields and text. In their model the generation is regarded as a sequence of decisions. Although the latter approach seems powerful for content selection, the fact that surface realisation is also performed jointly makes the approach difficult to be evaluated for content selection solely.

Konstas and Lapata (2012) present a framework for content planning by discriminatively re-ranking using the structured perceptron for learning. In this framework,

content features are seen as a hypergraph where nodes denote words. Although their system outperforms the one presented by Angeli et al. (2010), as mentioned before, evaluating only the content selection task is impossible. In addition, both systems are good in emulating the phenomena observed in the dataset, however the users' preferences / background / interests are not taken into account. We aim to fill this gap with the approaches we developed in Chapters 5 and 6.

Lampouras and Androutsopoulos (2013) present an Integer Linear Programming model for generation. Their model combines content selection, lexicalisation and sentence aggregation. The ultimate goal of this method is to produce compact text with as short length as possible given an entity of OWL ontology and a set of OWL axioms (facts).

More recently classifiers have been used for content selection to decide whether an element should be mentioned in the summary or not. Sowdaboina et al. (2014) use neural networks for segmentation which then influences the content selection decisions in the domain of weather forecasts. Kondadadi et al. (2013) report a statistical NLG framework for both content planning and realisation. Content is represented as semantic annotations and the realisation is performed using templates. The content selection and realisation decisions are learned from an aligned corpus using Support Vector Machines for modelling the generation and for creating a statistical model. Kondadadi et al. (2013) do not report using other algorithms for generation. In this thesis, Support Vector Machines are also used and compared to other classification algorithms, but it is found that in our domain, Decision Trees perform better, therefore they are used as a baseline system as we will discuss in Chapter 4.

For reasons already mentioned, direct comparisons are not applicable between the systems presented here. However, from the literature is evident that content selection approaches need two elements to work well:

- Content needs either to be considered collectively, or
- content selection decisions need to be made sequentially.

To emulate the two qualities discussed above, in this thesis, we formulate the task of content selection in two different ways: (1) we present an approach that treats content selection as a collective task (Section 4.2), and (2) we present an approach that treats content selection as a sequential task (Section 4.1). The approaches presented here are effective in their domains, therefore it is hard to argue that one approach is better than the others. Clearly, the NLG community needs data that are publicly available and in a ready-to-use form that can be used for generation tasks, as well as robust evaluation metrics.

Data-driven approaches can be developed quite fast and systems can be quickly re-trained on new datasets. However, not all approaches will work well on all datasets. One should initially perform a data analysis in order to understand the domain and then choose a content selection approach. The data should be checked for correlations, collinearities and other interactions. For instance, if events need to be mentioned together in order to make a coherent summary, one should choose an approach that considers data collectively. If what matters most is the sequence of the events, one should choose a method that selects the content in a sequential way. If there are no data interactions, one can opt for a simple method such as decision trees. Finally, one should consider the trade-offs between the development time arising from complex algorithms and the effectiveness of an approach.

The methods discussed in this section have not been used as part of a real system, therefore there is no evidence that these approaches fulfill task-based goals or address user preferences. They also lack in the user-adaptive quality compared to the rule-based systems. This thesis explores user-adaptation and makes significant contributions to the field of adaptive and non-adaptive NLG systems. Therefore, in the next section we discuss user-adaptive systems.

Author(s)	**User Model information**	Domain
Stock et al. (2007)	reason of visit, interest and change of background knowledge during the visit	museum
Demberg and Moore (2006)	user preferences on flights: price, number of stops, airport location etc.	flights
Williams and Reiter (2008)	users' readability skills	student feedback
Janarthanam and Lemon (2010)	user's inferred prior knowledge	instruction giving for internet connection setup
Han et al. (2014)	latent variables	rivers
Walker et al. (2007)	users' preferences	restaurant recommendations
Mairesse and Walker (2007)	Big Five Personality traits	personality recognition
Mahamood and Reiter (2011)	stress levels	health

Table 2.3: NLG systems that use User Models.

2.4 User-adaptive Systems

User-adaptation is understudied in the field of Natural Language Generation, although it is an area that has been studied in various fields of Computer Science such as Computer Human Interaction and in various commercial setups, such as Netflix's personalised movie rating prediction on. This section reviews current practices for user-adaptive output for data-to-text systems. One of the early approaches to adaptive Natural Language Generation is presented by Reiter et al. (1999) for the *STOP* system. This approach uses rules to map questionnaire answers to surface text. As each questionnaire only applies to a specific user, the output is personalised to a user's specific answers.

The predominant way of adaptive data-to-text system is through user modelling. Table 2.3 summarises NLG systems that use User Models (UMs) along with the information included in the UMs. In the context of museum exhibits, Stock et al. (2007) describe an adaptive multi-modal interactive system, *PEACH*, that is designed for mu-

seum visitors. It consists of:

- a virtual agent that assists visitors and attracts their attention,
- a user-adaptive video display on a mobile device, and
- a user-adaptive summary that is generated at the end of the visit in the museum.

A predecessor of PEACH is Ilex (O'donnell et al., 2001) which generates dynamic context in the domain of a virtual museum. PEACH is based on Ilex but it enhances the output with tailored video. In Ilex, the generation was tailored to user's specific attributes such as reason of visit, interest and change of background knowledge during the visit. The User Model is rule-based. Other innovations of Ilex include the fact that it allows the users to schedule their path through the museum and it makes use of the history to present richer summaries, for example by comparing different exhibits that the user has already visited. The content is selected by ranking content and the most relevant is selected.

Demberg and Moore (2006) also suggest a User Model approach to Information Presentation in the context of flight recommendation. In their approach, the selected content is influenced by the attributes a user finds important, such as price, number of stops etc. The novelty here is that, in order to increase user confidence, the attributes with low value in the User Model are briefly summarised, so as to help the user make an informed decision.

Williams and Reiter (2008) describe SkillSum, a system that generates personalised feedback report for someone that just completed a test on numeracy and literacy. The reports are personalised to the users' readability skills and is developed in a rule-based fashion. The level of readability is derived by the users' answers to the test. The participants significantly preferred the personalised feedback to canned output.

NLG systems have also used User Models in order to adapt their linguistic output to individual users (Janarthanam and Lemon, 2010; Thompson et al., 2004; Zukerman and Litman, 2001). For instance, Janarthanam and Lemon (2010) propose a system that

adapts the generated referring expressions to the user's inferred prior knowledge of the domain. As a user's prior knowledge can change through interactions, they introduce *dynamic user modelling* which allows for updates to the User Model after interacting with the user.

Han et al. (2014) suggest the use of latent User Models to NLG. In this framework, instead of directly seeking the users' preferences or the users' knowledge through questionnaires, the UMs are inferred through "hidden" information derived from sources such as *Google Analytics*.

Walker et al. (2007) present an approach that adapts its surface realisations to individual users' preferences, by using a generate-and-rank approach. The ranking step of this approach is influenced by the individual's own preferences and therefore the generated realisations are different for each user. Mairesse and Walker (2007) present a system that recognises the *Big Five* personality traits and use this information for adapting the surface text to a particular user's personality.

NLG systems can employ different versions of a system for each different user group (Gatt et al., 2009; Hunter et al., 2011; Mahamood and Reiter, 2011). The BT project uses NLG systems in a Neonatal Intensive Care Unit environment to automatically provide reports to different stakeholders. For example, BT-nurse is addressed to nurses working in NICU whereas BT-family is addressed to the parents and relatives of the baby and is able to further adapt to users' stress levels.

Dethlefs et al. (2014) move away from user modelling by exploiting user ratings to infer users' preferences on utterances describing restaurant suggestions. In this setup, users are clustered in terms of linguistic preferences. Then, users prospective ratings can be predicted by assigning a user in a cluster and by averaging over ratings of other users in the same cluster. In Dethlefs et al. (2014) setup, the ratings have been derived by users. In specialised domains, ratings should be gathered not only by users, but also by experts, as for instance in the student feedback generation domain. Lecturers (i.e. experts) are responsible for producing feedback for students, whereas students

are the receivers of the feedback. In this case, clustering users in terms of preferences would not be appropriate, as students do not know how to effectively produce feedback. Therefore, in this thesis we deal with the challenge of finding middle ground between the preferences of "speakers" and "hearers", when these two types of stakeholders are known and we contribute a methodology that addresses this challenge (Chapter 5). In addition, we extend the approach presented by Dethlefs et al. (2014) to handle first-time users, without needing to acquire user ratings (Chapter 6).

2.5 Evaluation Methods

In this section, we initially refer to different evaluation methods for NLG systems and we discuss which methods we use in this thesis. Evaluation of a user tailored system is important in order to improve user experience (satisfaction), task efficiency and task effectiveness. In the next section, we review some forms of evaluation noting any relevance with the related work mentioned earlier. The evaluation methods are categorised into intrinsic and extrinsic methods, adopting the terminology used by Belz and Hastie (2014).

2.5.1 Intrinsic Evaluation

Intrinsic evaluation methods are useful for performing fast, preliminary comparisons and benefit from not needing human participants. Intrinsic methods can be split into (1) output quality measures and (2) user like ratings. Both are discussed below.

2.5.1.1 Output Quality Measures

Automatic metrics are a type of intrinsic evaluation which assess the similarity of the output to a reference model or assess quality criteria (Belz and Hastie, 2014), such as the translation metrics BLEU, NIST, ROUGE, F-measure etc.

- BLEU (Bilingual Evaluation Understudy) evaluates the output quality of machine

translated text by comparing the machine translated text to a human reference translation, so that "the closer a machine translation is to a professional human translation, the better it is" (Papineni et al., 2002). It can be also applied to generation systems to measure the proximity of a machine generated text to a human generated text, as used by Angeli et al. (2010).

- NIST (named after the US National Institute of Standards and Technology) is based on BLEU but it also assesses how informative an n-gram is by scoring high for rarer n-gram occurrences (Doddington, 2002).

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation) package is used both in machine translation and in summarisation evaluation. It compares the output text against a reference text (Lin, 2004). ROUGE is a summarisation evaluation package which consists of several automatic metrics: 1) ROUGE-N, which is based on n-grams, 2) ROUGE-L, which is based on Longest Common Subsequence, 3) ROUGE-W, which is based on Weighted Longest Common Subsequence and 4) ROUGE-S, which measures the overlap of skip-bigrams between a generated summary and a reference summary.

- F-measure is borrowed from statistics and is based on precision and recall (Olson and Delen, 2008). This measure can evaluate the content selection as discussed by Angeli et al. (2010) and in Chapter 4.

Another way of intrinsic evaluation is human-assessed evaluation, where humans evaluate the generated output in terms of similarity to a reference summary/translation as described by Belz and Kow (2010). In human-aided Machine Translation, post-editing is used to improve the output after machine translation and thus the generated output can be evaluated (Hutchins and Somers, 1992). This metric has been used for natural language generation too, as for instance in (Sripada et al., 2005).

In this thesis, automatic metrics have been used such as F-score, precision, recall and BLEU. Automatic metrics are regarded as "backup" metrics and they are used in

conjunction with human evaluations. They are not standalone metrics and their results are not always correlated with human evaluations (Belz and Reiter, 2006). The results of a human evaluation are more important, because what really matters is the usability of a system, therefore human evaluations are of high importance and have been used for all experimental setups in this thesis. In particular, a user-like measure has been used and it is described in the next section.

2.5.1.2 User-like Measures

User like measures are used to assess the systems' output or a particular module. For this evaluation, users are asked questions such as "How useful did you find the summary?" (Belz and Hastie, 2014). This kind of method is used by Walker et al. (2002) and Foster and Oberlander (2007), where an adaptive system is compared to a non-adaptive.

2.5.2 Extrinsic Evaluation

Extrinsic evaluation methods are useful for defining what an application is good for and to identify whether an application fulfils its task requirements. The extrinsic metrics can be split into (1) user task success and (2) system purpose success. Both are discussed in the next sections.

2.5.2.1 User Task Success Measures

User task success measures measure anything that has to do with what the user gains with the systems' output, such as decision making, comprehension accuracy etc. (Belz and Hastie, 2014). Such an evaluation is used in BabyTalk (Gatt et al., 2009), where the users are shown two outputs and have to make a decision, so as to measure which output is more efficient and helpful in decision making.

2.5.2.2 System Purpose Success Measures

System purpose success measures evaluate a system by measuring whether it can fulfil its initial purpose (Belz and Hastie, 2014). Such an evaluation is applied to the STOP system (Reiter et al., 1999) in order to find out whether the purpose of the system was achieved, i.e. to define whether users quit smoking.

Although extrinsic evaluation is extremely important, it is an expensive and time consuming task, as users need to be recruited. In addition, it may be uncertain whether the system is solely successful or whether there are external factors that influence the outcome. For example, if we consider the STOP project, there is uncertainty of whether someone quitted smoking because of the generated letter or due to other circumstances, such as health issues.

2.6 Conclusions

This chapter introduced the state-of-art data-to-text system architecture and described the main distinct components of such systems. It reported the two main approaches to data-to-text generation: rule-based approaches and data-driven approaches. It introduced content selection in other domains and it reviewed adaptive NLG systems. Finally, it discussed evaluation metrics and their suitability.

Both rule-based and data-driven approaches provide benefits and suffer from limitations as we will discuss in the following paragraphs and as it is depicted in Table 2.4. Regarding content selection, rule-based systems based on crafted rules, corpus analysis and expert consultations (Knowledge Acquisition from experts) are robust and widely used in industry. For surface realisation, the output produced by rule-based systems is more understandable by humans, with no ungrammatical elements as it is fully controlled. Rule-based systems can also account for outliers as long as enough rules have been provided to handle extreme examples of data. However, they may not be able to cover all distinct cases as the number of rules increases proportionally to

Approaches	Strengths	Limitations
Rule-based	- robust in small domains - understandable output - thoroughly studied - suitable for commercial use	- expert knowledge required, expensive - not transferable - number of rules increases analogously to the domain complexity
Data-driven	- cheap, fast to be implemented - scalable - methods can be reused for new domains - experts are not always required - can make inferences from data - are flexible in accounting for user preferences	- can produce non-understandable output - require large datasets - systems reflect the quality of the data

Table 2.4: Strengths and limitations of the two approaches to data-to-text systems

the complexity of the domain. In addition, the cost of developing and maintaining a rule-based system is high compared to systems that use data-driven approaches, as the latter can be scalable by providing more data. In addition, rules are domain specific and therefore not easily transferable to other domains.

Statistical methods and Machine Learning approaches, have been widely used and adopted in other NLP systems as compared to data-to-text systems. With statistical methods, NLG systems have the potential to accommodate adaptation (Lemon, 2011), be more domain independent, automatically optimised and generalised (Angeli et al., 2010; Rieser et al., 2010; Dethlefs and Cuayahuitl, 2011). Content selection algorithms designed for learning from data corpora can be more efficient, easily ported in new applications and cheap. Due to their ability to take into account large corpora, their coverage can be extended by using more training examples. Statistical methods can be more expressive than rule-based systems in many ways, linguistically and adaptively and offer scalability and flexibility. Data-driven methods can be implemented faster than rule-based system through code reusing and the use of off-the-shelf tools. Finally, data-driven approaches can work well with small datasets, as they have the ability to generalise for unseen scenarios. This is an important quality if one considers the time

and resources needed for data collection and corpus creation. The utmost advantage of machine learning methods is that they can make inferences when humans cannot. This quality can assist in developing more informed systems. Also statistical methods do not require the acquisition of knowledge from experts who can be hard to recruit. Finally, data-driven methods can be more flexible in accounting for user preferences. However, data-driven approaches require large amounts of data for training and they are sensitive to data quality (Reiter et al., 2003).

For the aforementioned reasons, this thesis will be concerned with the development of data-driven approaches to content selection with a focus on user adaptation. These approaches will be compared with meaningful baselines, such as a rule-based system designed with the assistance of an expert, user-adaptive systems etc. We will evaluate these approaches using a combination of automatic metrics and user-like measures.

Chapter 3

Framework and Data

This chapter presents the Natural Language Generation (NLG) framework developed throughout the thesis and the two data collections performed. The chapter consists of four parts: (1) we describe the nature of the task at hand (Section 3.1); (2) we describe the overall NLG setup and we pinpoint the content selection module within the NLG framework (Section 3.2). We then describe the two experimental domains along with two data collections we performed for the purposes of this research, namely: (3) student feedback generation (Section 3.3) and (4) health informatics (Section 3.4).

3.1 The Task: Content Selection in Data-to-text Systems

The task of content selection is formulated as follows: given the time-series data, determine the appropriate content to be selected. Content selection decisions based on trends in time-series data determine the selection of the useful and important time-series variables that should be conveyed in the summary. We refer to the time-series variables as *factors* in both domains. The decision of factor selection can be influenced by other factors whose values they correlate with. They can be also based on the appearance or absence of other factors in the summary. Finally, they can be based on

the factors' behaviour over time. Moreover, some factors may have to be discussed together in order to achieve some communicative goal, for instance, a teacher might want to refer to student's marks as a motivation for increasing the number of hours the student studies.

3.2 Overview of NLG Framework

This section describes the NLG architecture (Figure 3.1) that has been designed for this thesis. It includes the following modules: (1) a data analysis module (Section 3.3.2) which is responsible for translating the time-series data into trends or identifying useful events; (2) a content selection module (Chapters 4, 5 and 6); and (3) a template-based surface realiser (Section 3.3.3 and 3.4.2). In this chapter, (1) and (3) are only described, as these will remain the same throughout the thesis.

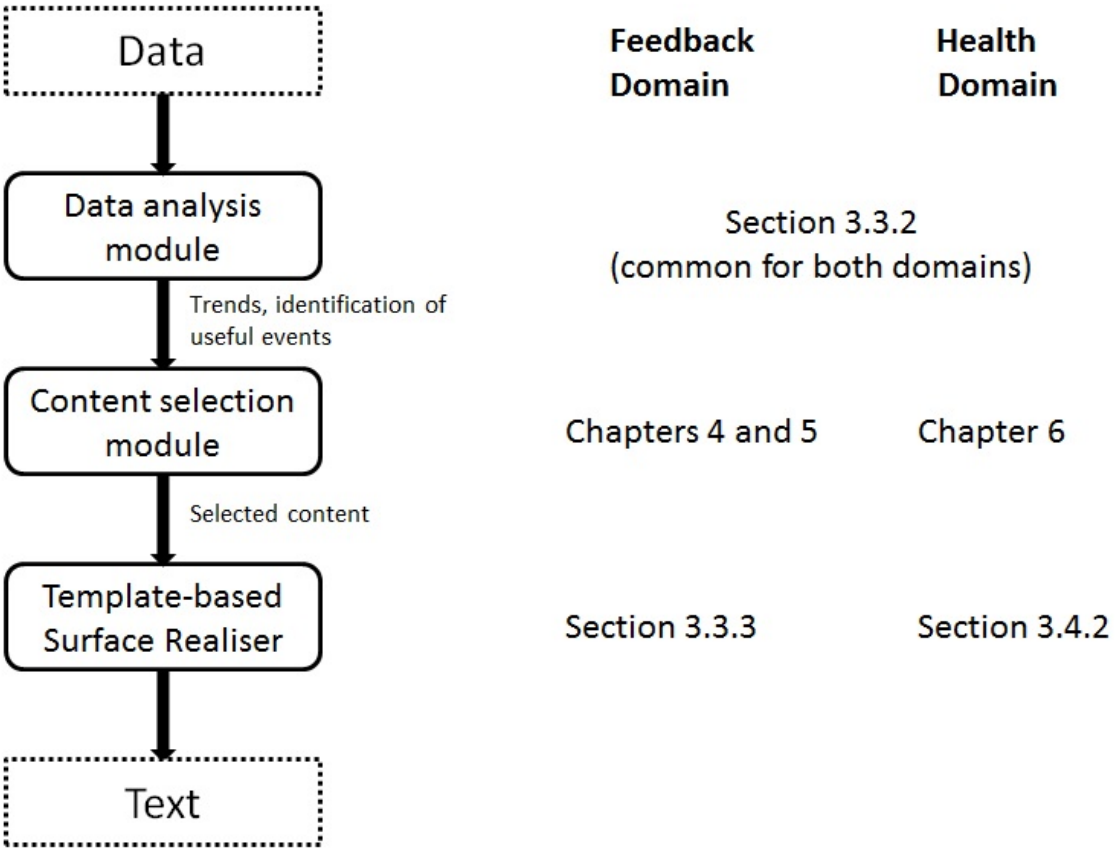


Figure 3.1: Data-to-text system architecture.

Our architecture differs from the one proposed by Reiter and Dale (2000) as we describe in this section. Our domains do not use event input data, therefore a data interpretation module is not required in the same sense as in Reiter's architecture. We developed a *data analysis* module which has characteristics from the first two of Reiter's suggested modules (signal analysis and data interpretation). It is responsible for handling the input raw data, identifying trends, estimating the average and finding highs and lows. The goal in our domains is to generate paragraph-long summaries. Therefore, there is no need for a document planning module, as there are no decisions made on document structure such as paragraph breaks. Therefore, we developed a content selection module that is responsible only to choose the information to be conveyed and the ordering of the information.

3.3 Domain: Student Feedback

Data collection and corpus construction are critical processes for developing NLG systems in domains when no previous corpora exist. The first domain we address is student feedback generation. To our knowledge, this is the first effort to generate student feedback in terms of a real-world classroom, rather than in terms of an online tutoring system. However, there is previous work on modelling tutor's feedback in online environments, as described by Porayska-Pomsta and Mellish (2013) and Moore et al. (2004).

In the following sections, we initially discuss the factors that influence students' learning and then we describe our data collection with students. Figure 3.2 shows the corpus creation steps. In Section 3.3.2, we present an overview of the dataset collected, and then we describe how we process the collected time-series data and how we exploit the information in order to create relevant templates by working with an expert. Finally, we conduct an additional data collection with lecturers in order to create an aligned dataset (student data and feedback summaries).

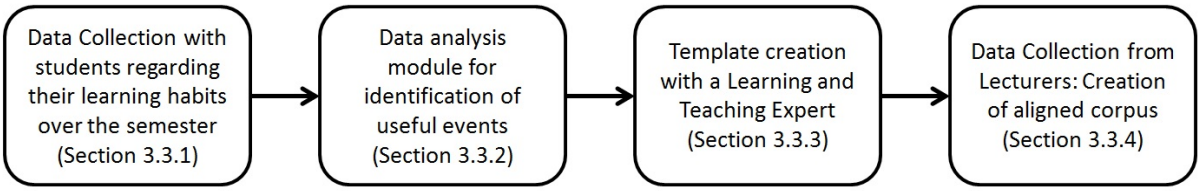


Figure 3.2: Corpus creation work-flow in student feedback domain.

3.3.1 Data Collection from Students

Twenty-six MSc and undergraduate students were recruited who attended the module of Artificial Intelligence at Heriot-Watt University in 2012. They were asked to fill in a web-based diary-like questionnaire during the lab sessions of this module which was taught over the course of a 10 week semester. Initially, the students were asked to provide some demographic details: (1) age; (2) gender; (3) place of birth; (4) native-English speaker; and (5) BSc/MSc student. The ages of the participants range between 19 to 29. There are 15 native-English and 11 non-native speakers, from 7 countries (China, France, Germany, Greece, Nigeria, UK, US).

In addition, students provided information on a weekly basis for eight factors that could influence their performance. We also included as a ninth factor the marks achieved by students each week. These nine factors (including marks) were motivated from the literature and are listed here in terms of `effort` (Ames, 1992), `frustration` (Craig et al., 2004), `difficulty` (Person et al., 1995; Fox, 1993) and `performance` (Chi et al., 2001). `Effort` is measured by three factors: (1) how many hours they studied; (2) the level of revision they have done; (3) as well as the number of lectures (of this module) they attended. `Frustration` is measured by (4) the level of understandability of the content; (5) whether they have had other deadlines; and whether they faced any (6) health and/or (7) personal issues and at what severity. The `difficulty` of the lab exercises is measured by (8) the students' perception of difficulty. Difficulty and understandability describe different states. Difficulty refers to the material, whereas understandability refers to the students' ability to learn the material. Very difficult material can result in students' high understandability if for instance the students

dedicate many hours studying. Easy material might not be understood by the students in case they miss classes. Finally, (9) marks achieved by the students in each weekly lab was used as a measure of their `performance`. The questions asked can be seen in Table 3.1 and an example of the data collected by one student is shown in Figure 3.2. An analysis of the dataset collected is shown in Table 3.3. The min, max and mode show that there is variability in the collected data. The standard deviation quantifies the variation of the data. We can observe that high variation exists mainly in `marks`.

Factors	**Questions**	**Potential answers**
1. Hours studied	How many hours did you spend on studying for this module this week?	1-5
2. Understandability	How would you rank your understandability of this week's material?	1-5
3. Difficulty	How would you rank the level of difficulty of the lab exercises ?	1-5
4. Deadlines	Are there any deadlines for other modules this week?	1-5
5. Health issues	Do you currently suffer from any health condition?	1-5
6. Personal issues	Any other personal issues going on?	1-5
7. Lectures attended	How many lectures of the Artificial Intelligence module are you intending to attend this week?	0-3
8. Revision	Have you revised the material given so far for this module?	1-5

Table 3.1: The questions regarding the learning habits.

3.3.2 Data Analysis Module

The data analysis module is responsible for analysing each student's time-series data. Initially, the data are processed so as to identify the existing trends of each factor during the semester (e.g. number of lectures attending decreases). The tendencies of the data are estimated using linear regression, with each factor annotated as INCREASING, DECREASING or OTHER. There are cases where the data do not disclose a clear tendency. For instance, a student can initially have an increasing performance and

Week	hours studied	understandability	difficulty	dead-lines	health issues	personal issues	lectures attended	revision	marks
2	3	3	3	1	1	1	3	2	5
3	2	4	3	1	1	1	2	2	4
4	1	2	2	2	2	1	1	3	0
5	1	1	3	2	2	1	1	3	0
6	1	1	4	4	2	1	2	3	0
7	1	1	4	5	1	1	1	3	0
8	1	1	4	3	1	1	2	3	0
9	1	1	3	4	1	1	1	3	0
10	1	1	3	4	1	1	2	3	0

Table 3.2: Example time-series information from one student. The data correspond to the answers given to questions in Table 3.1 by the student.

	hours studied	understandability	difficulty	dead-lines	health issues	personal issues	lectures attended	revision	marks
min	1	1	1	1	1	1	0	1	0
max	5	5	5	5	5	5	3	5	5
mode	1	3	3	1	1	1	3	3	0
sd	1.035	1.005	0.878	1.334	0.73	1.068	0.7	0.679	2.399

Table 3.3: Min, max, mode and standard deviation of the dataset. For marks, there were 106 instances of 0 and 104 instances of 5 (mean = 2.55).

then decreasing. The module is able to identify these segments, i.e. the trend changes, and map them to relevant templates (as we will discuss in Section 3.3.3). In the case of segments, the start and end points do not correspond to the initial value, for instance in week 1, as the segment might describe the weeks 3 to 9. Therefore, we use the first and last value of the time-series to describe the data, for instance, "Your marks decreased from 5 to 3.", where 5 is the initial measurement and 3 the final. Secondly, for each student a comparison between their average of each factor and the class average of the same factor is automatically performed. Thirdly, the data analysis module is able to identify unusual or notable events given predefined thresholds. For instance, it is able to identify which weeks a student faced health issues or scored 0 at the lab exercises.

In the collected data, lecturers did not comment on different segments, therefore, the templates included for realisation do not describe the different segments. In addition, OTHER is also used to describe factors when they are stable. Lecturers do not comment on stable factors, but instead they prefer to make general statements that can be more useful. For instance, if a student achieved high marks in all lab sessions, it is preferred to mention "Keep up the good work!", instead of "Your performance was stable!".

3.3.3 Template Creation

The wording and phrasing used in the templates to describe the data were derived from working with and following the advice of a Learning and Teaching (L&T) expert. The expert provided consultation on how to summarise the data. We derived four different types of templates for each factor: <trend>, <average>, <weeks> and <other> based on time-series data on plotted graphs and the data analysis module described above. The templates describe these factors in four different ways:

1. **<trend>:** referring to the trend of a factor over the semester, or the changes of the trend during the semester (e.g. "Your performance **was increasing**..." or "You performed better in the first half of the semester comparing to the second half.."),

2. **<average>:** considering the average of a factor, or by comparing the student's average with the class average value (e.g. "You dedicated **1.5 hours studying on average**..."),

3. **<weeks>:** explicitly describing the value of the factor at specific weeks (e.g. "In **weeks 2, 3 and 9**...") and

4. **<other>**[2]**:** mentioning other relevant information (e.g. "**Revising material will improve your performance**").

[2]<other>as a template is different from the value OTHER of the template <trend>.

The exhaustive list of the templates is shown in Appendix A. For each student, 28 templates are generated that describe the time-series data[3].

In addition, the L&T expert consulted on how to enhance the templates so that they are appropriate for communicating feedback according to the guidelines of the Higher Education Academy (HEA, 2009), for instance, by including motivating phrases such as "You may want to plan your study and work ahead".

3.3.4 Data Collection from Lecturers

In order to create an aligned corpus that can be used for training machine learning algorithms, we collected data from lecturers. 11 lecturers selected the content to be conveyed in a summary, given the set of raw data. The data collection consisted of three stages where lecturers were given plotted factor graphs and were asked to:

1. write a free style text summary for three students (Figure 3.3),
2. construct feedback summaries using the templates for three students (Figure 3.4), and
3. rate random feedback summaries for two students (Figure 3.5).

The data collection was developed using the Google Web Toolkit[4] for Web Applications, which facilitates the development of client-server applications. The server side hosts the designed tasks and stores the results in a datastore. The client side is responsible for displaying the tasks on the user's browser. For all tasks, the factor data are plotted in separate graphs, using HighCharts[5].

In Task 1, the lecturers were presented with the factor graphs of a student (one graph per factor) and were asked to provide a free-text feedback summary for this student. The lecturers were encouraged to pick as many factors as they find useful and to discuss the factors in any order they find appropriate. Figure 3.3 shows an example

[3]There were not $4x9 = 36$ as some template types were not available for all factors, e.g. <other>
[4]https://code.google.com/p/google-web-toolkit/
[5]http://www.highcharts.com

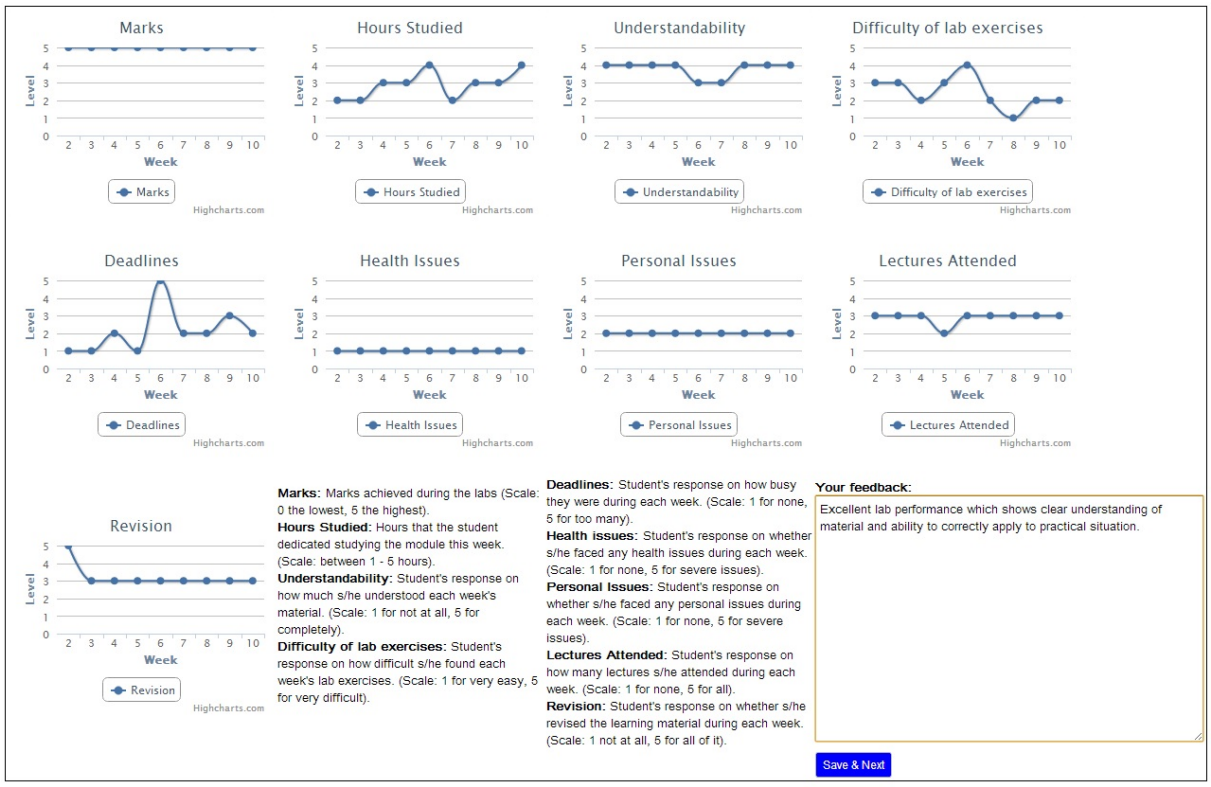


Figure 3.3: The interface of the 1st task of the data collection: the lecturer consults the factor graphs and provides feedback in a free text format.

free text summary for a highly performing student where the lecturer decides to talk about lab marks and understandability. Each lecturer was asked to repeat this task 3 times for 3 randomly picked students.

In Task 2, the lecturers were again asked to construct a feedback summary but this time they were given a range of sentences generated from the templates. They were asked to use these templates to construct a feedback report. The number of alternative utterances generated for each factor varies depending on the factor and the given data. In some cases, a factor can have 2 generated utterances and in other cases up to 5 (with a mean of 3 for each factor) and they differentiate in the style of trend description and wording. Again the lecturers were asked to choose which factors to talk about and in which order according to their preferences, as well as to decide on the template style he/she would prefer for the realisation through the template options. Figure 3.4 shows an example of template selection for the same student as in Figure 3.3.

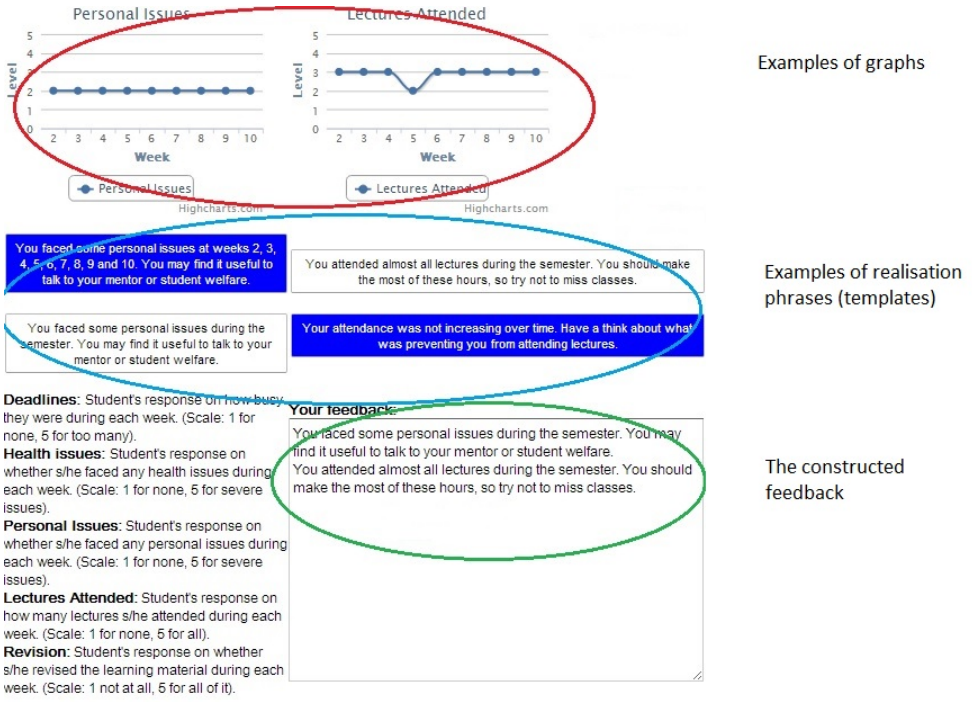


Figure 3.4: The interface of the 2nd task of data collection: the lecturer consults the graphs and constructs a feedback summary from the given templates (this graph refers to the same student as Figure 3.3).

In Task 3, the lecturers were presented with the plotted factor graphs plus a corresponding feedback summary that was generated by randomly choosing n factors and their templates, and were asked to rate it in a scale between 0 - 100 (100 for the best summary). Figure 3.5 shows an example of a randomly generated summary for the same student as in Figure 3.3.

Factor	**(1) M**	**(2) HS**	**(3) Und**	**(4) Diff**	**(5) DL**	**(6) HI**	**(7) PI**	**(8) LA**	**(9) R**
(1) M	**1***	**0.52***	**0.44***	**-0.53***	-0.31	-0.30	**-0.36***	**0.44***	0.16
(2) HS	**0.52***	**1***	0.23	-0.09	-0.11	0.11	-0.29	0.32	**0.47***
(3) Und	**0.44***	0.23	**1***	**-0.54***	0.03	-0.26	0.12	**0.60***	0.32
(4) Diff	**-0.53***	-0.09	**-0.54***	**1***	0.16	-0.06	0.03	-0.19	0.14
(5) DL	-0.31	-0.11	0.03	0.16	**1***	0.26	0.24	**-0.44***	0.14
(6) HI	-0.30	-0.11	-0.26	-0.06	0.26	**1***	0.27	**-0.50***	0.15
(7) PI	**-0.36***	-0.29	0.12	0.03	0.24	0.27	**1***	**-0.46***	**0.34***
(8) LA	**0.44***	0.32	**0.60***	-0.19	**-0.44***	**-0.50***	**-0.46***	**1***	-0.12
(9) R	0.16	**0.47***	0.03	0.14	0.14	0.15	**0.34***	-0.12	**1***

Table 3.4: The Pearson correlation coefficients of the data attributes (* means $p < 0.05$).

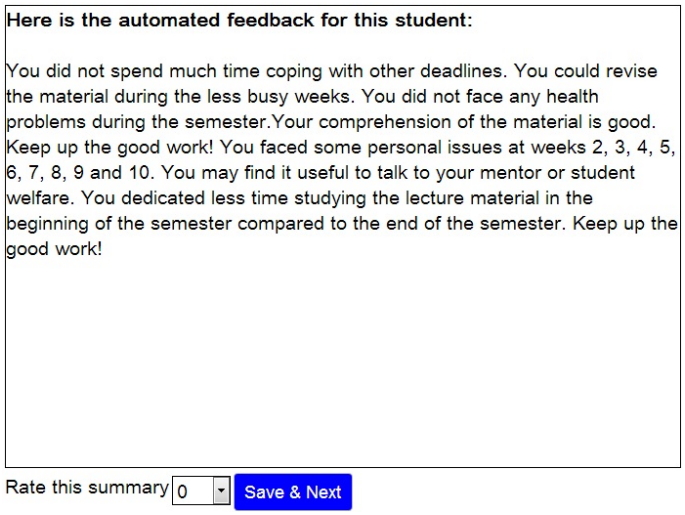


Figure 3.5: The interface of the 3rd task of data collection: the lecturer consults the graphs and rates the randomly generated feedback summary (this graph refers to the same student as Figures 3.3 and 3.4).

Finally, our analysis of the collected data shows that there are significant correlations between the factors. For example, the number of lectures attended (LA) correlates with the student's understanding of the material (Und), $r = 0,60$. As expected, marks are correlated with almost every other factor, apart from the deadlines ($r = -0.31$) and the revision ($r = 0.16$), but they are strongly correlated with the hours studied ($r = 0.52$) and the difficulty of the learning material ($r = -0.53$). Surprisingly, the number of hours a student studied do not correlate with understandability of the material ($r = 0.23$), difficulty ($r = -0.09$) and other deadlines ($r = -0.11$). For a more thorough view on the correlations please see Table 3.4.

3.3.5 Discussion

The analysis reveals that there are correlations between the learning factors which influence students' performance. These correlations should be accounted for. Indeed, they guide our algorithm selection. In Section 4.2, we describe a multi-label classification approach to content selection, which is very efficient because it takes into account the correlations of the available content when making generation decisions.

3.4 Domain: Health Informatics

The second domain we address is health informatics. The data collection is based on scenarios provided by the Managing Information in Medical Emergencies - MIME[6] project from University of Aberdeen. Although MIME's aim is to automatically produce handover reports, here we investigate how to better communicate sensor data based on these scenarios. The aim of the data collection is, therefore, to create an aligned corpus, which allows us to study and define the preferences of different users. Similar to the feedback generation domain, the data collection follows the pipeline shown in Figure 3.6.

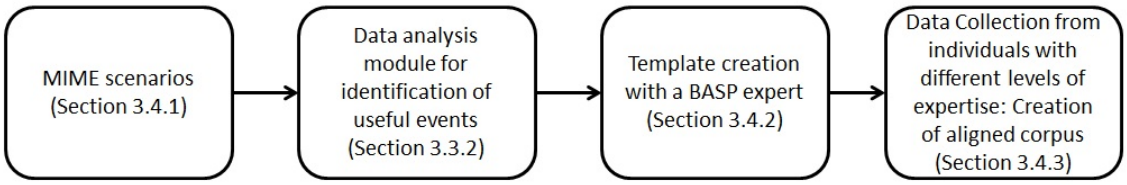


Figure 3.6: Corpus creation work-flow in health informatics domain.

3.4.1 The MIME Scenarios

From the MIME project, we used four scenarios that describe medical emergencies. Each scenario consists of a textual description of the incident (e.g. see top of Figure 3.7), three graphs that correspond to the physiological measurements of Breathing Rate (BR), Blood Oxygen Saturation (SpO_2) and Heart Rate (HR). All 4 scenarios can be found in Appendix B. A description of each scenario is shown in Table 3.5.

3.4.2 Template Creation

We work with a medical expert (a first aid trainer) in order to identify six potential textual descriptions/phrase templates of describing time-series data:

[6]http://www.dotrural.ac.uk/mime/

Scenario	Brief description
1. Smoke inhalation	The patient inhaled smoke through a building fire
2. Drowning	The patient swallowed water after falling into water
3. Falling down stairs	The patient fell off the stairs
4. Bicycle accident	A cyclist was hit by a car

Table 3.5: A brief description of each scenario.

1. **<average>:** considering the average of a factor (e.g. "The heart rate was 114 beats per minute"),

2. **<trend-verbose>:** referring to the trend of a factor in a verbose way (e.g. "The heart rate increased from 110 to 121 beats per minute"),

3. **<trend-succinct>:** referring to the trend of a factor in a succinct way (e.g. "Heart rate *uparrow* from 110 to 121"),

4. **<range-verbose>:** referring to the range of values observed in a verbose way (e.g. "The heart rate was between 108 and 121 beats per minute"),

5. **<range-succinct>:** referring to the range of values observed in a succinct way (e.g. "Heart rate 108-121") and

6. **<inference>:** making an inference from the data (e.g. "Heart rate observation problematic").

3.4.3 Corpus Creation

We initially describe the data collection and next we analyse user preferences when describing sensor data that measure the three physiological conditions: Breathing Rate, Blood Oxygen Saturation (SpO2) and Heart Rate. In the processed time-series data, each time-stamp corresponds to one minute of measurements. For each graph, there are six ways of referring to the measured parameter as described previously. Each participant is asked to choose the phrase that s/he would use to describe each condition.

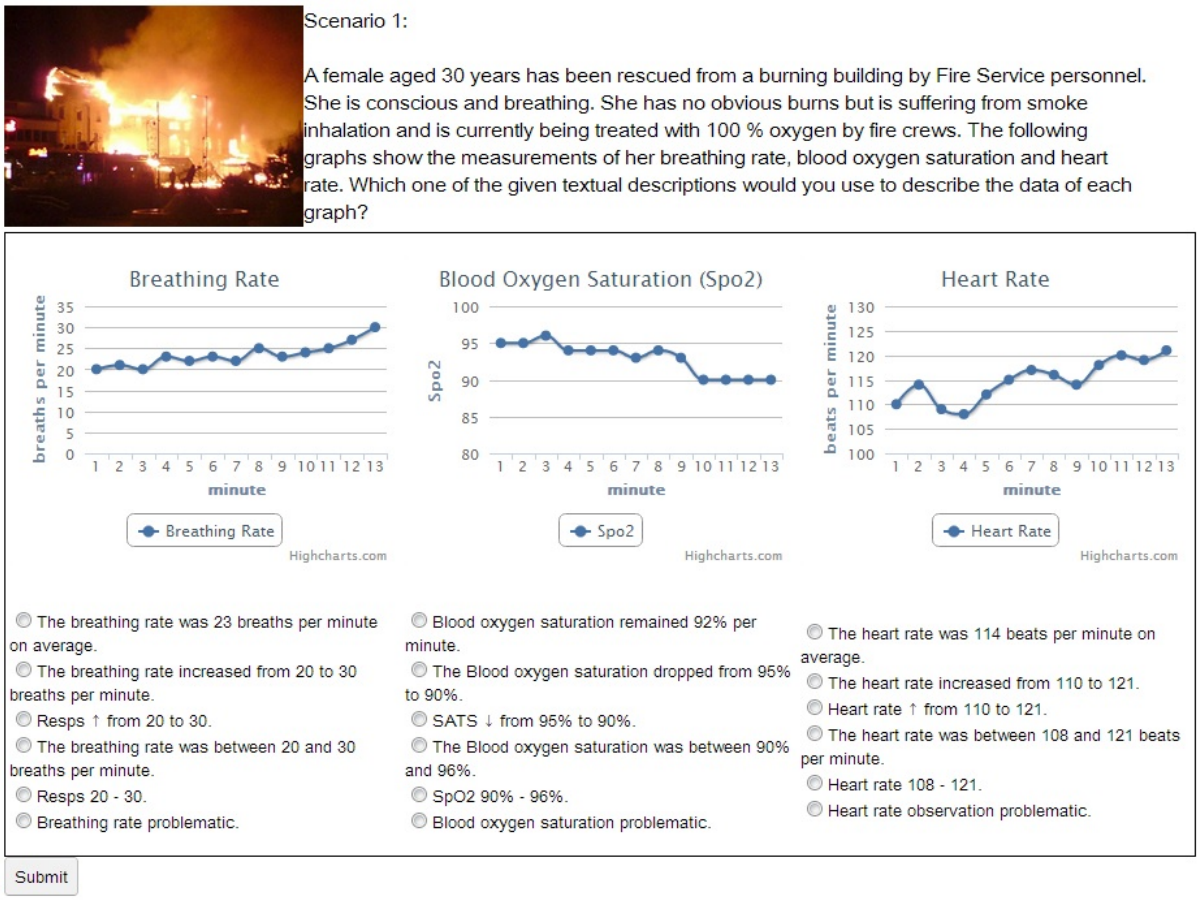


Figure 3.7: In the top, a textual description of the event is given. In the middle, the graphs present the processed physiological data and in the bottom six different phrases that describe each parameter is given.

We conduct the data collection online using the Google Web Toolkit. We recruited sixty-nine people via email (34 males and 35 females). Their background varies, from people who have not received any medical training to medical doctors. The participants are asked to assign themselves to one of six possible groups regarding their training level in pre-hospital care, as presented in Table 3.6. Note that none of the participants classified themselves as members of Group 4 (Emergency Medical Technician, Ambulance Technician, Combat Medical Technician 1, Offshore Medic). They are also asked for additional information: gender; previous experience with sensor data; profession.

Each participant was shown four different emergency scenarios. Figure 3.7 shows an example scenario, the graphs that depict the sensor data and six corresponding textual summaries for each graph. Note that the order of the sentence in Figure 3.7 corresponds

Group	Level of training	Number of participants
Group 1	None	10
Group 2	First Aid at Work, Emergency	42
Group 3	Basic First Person on Scene, Intermediate First Person on Scene, Equivalent to BASP Advanced First Aid, Combat Medical Technician 2	5
Group 4	Emergency Medical Technician, Ambulance Technician, Combat Medical Technician 1, Offshore Medic	0
Group 5	Paramedic, Nurse, Physician's Assistant	4
Group 6	Medical Doctor	8

Table 3.6: The different levels of pre-hospital training

to the template types defined above. The templates are also listed in the pictures in Appendix B.

3.4.4 Corpus Analysis

The data analysis reveals a number of interesting results regarding the user preferences. Generally the phrase choices are influenced not only by the training level / occupation but also by the incident scenario they describe. In the following sections, we analyse observed differences in the data with a Pearson's Chi squared test.

3.4.4.1 Scenario / Phrase Choice Relation

The data reveals that the phrase choice is correlated with two factors: the scenario and the physiological condition (i.e. Breathing Rate, Blood Oxygen Saturation and Heart Rate). For a detailed description of the frequencies please see Table 3.7. For example, for the *smoke inhalation scenario*, we find that 66% of the participants choose to describe the Breathing Rate by mentioning the trend (increase) in a verbose way (template 2), whereas only 22% of the participants would describe the breathing rate using the succinct phrase of mentioning the trend (template 3). A similar distribution

Scenario	Template	Breathing rate	SpO2	Heart rate
1. Smoke inhalation	(1) Average	12.9%	0%	1.2%
BR: incr	(2) Trend verbose	66.2%	63%	46.7%
SpO2: decr	(3) Trend succinct	22%	28.5%	26.3%
HR: incr	(4) Range verbose	1.2%	3.8%	3.8%
	(5) Range succinct	6.4%	0%	0%
	(6) Inference	2.5%	3.8%	3.8%
2. Drowning	(1) Average	32%	35.2%	0%
BR: stable	(2) Trend verbose	1.4%	0%	61.2%
SpO2: stable	(3) Trend succinct	1.4%	4.4%	32.2%
HR: incr	(4) Range verbose	30%	44%	0%
	(5) Range succinct	17%	14.7%	1.4%
	(6) Inference	16%	1.4%	7.3%
3. Falling down stairs	(1) Average	19%	25.3%	0%
BR: stable	(2) Trend verbose	23%	1.5%	60.3%
SpO2: stable	(3) Trend succinct	15%	4.7%	33.3%
HR: decr	(4) Range verbose	17%	47%	3.1%
	(5) Range succinct	9.5%	19%	1.5%
	(6) Inference	14%	1.5%	1.5%
4. Bicycle accident	(1) Average	6.4%	30.6%	0%
BR: incr	(2) Trend verbose	51%	1.6%	61.2%
SpO2: stable	(3) Trend succinct	25%	9.6%	32.2%
HR: incr	(4) Range verbose	12%	41.9%	0%
	(5) Range succinct	3.2%	16.1%	1.6%
	(6) Inference	0%	0%	4.8%

Table 3.7: The phrase frequencies (%) of each scenario.

can be observed for the phrases chosen for the Blood Oxygen saturation variable: 63% of the participants chose template 2 and 28% template 3. For describing Heart Rate data, 46% of the participants choose the verbose way of describing the trend (template 2), whereas 36% would choose the succinct way (template 3). In sum, participants' choices mainly vary between template 2 and 3 and these choices are significantly different for different types of physiological data.

For the *drowning scenario*, 32% of the users prefer to refer to Breathing Rate by mentioning the average (template 1) and 30% prefer to mention the range in a verbose way (template 2). This is quite different from the observations derived from the smoke

inhalation scenario. The results differ for the Blood Oxygen Saturation as well. 35.2% prefer to refer to it by mentioning the average (template 1), whereas 44% prefer to mention the range in a verbose way (template 4). Finally, 93.4% of the users prefer to mention the trend when referring to the Heart Rate (61.2% by mentioning it in a verbose way (template 2) and 32.2% by mentioning it in a succinct way (template 3).

For the *fall down stairs scenario*, the users' preferences on mentioning the Breathing Rate are spread around the six templates. On the other hand, users prefer to mention the range of the Blood Oxygen Saturation in a verbose way (template 4, 47%) or mention the average (template 1, 25.3%). Regarding the Heart Rate, 93.6% of the participants prefer to mention the trend in a verbose (template 2, 60.3) and in a succinct way (template 3, 33.3%).

Finally, for the bicycle accident scenario, 51% of the users prefer to talk about the Breathing Rate by mentioning the trend in a verbose way (template 2), whereas 25% of the users prefer the succinct way of referring to the trend (template 3). For the Blood Oxygen Saturation, the results are similar to the drowning scenario. 30.6% of the users prefer the average template (template 1) and 41.9% of the users prefer to refer to range in a verbose way (template 4). Finally, the preferences for the Heart Rate remain similar, with 61.2% of the users preferring verbose way of referring to trend (template 2) and 32.2% preferring the succinct way (template 3).

Generally, it is observed that users select templates depending not only on the physiological data but their choices are also dependent on the scenario. However, there is no clear evidence on what drives the differences between the template selections.

3.4.4.2 Training Level / Phrase Choice Relation

As previously mentioned, the participants are associated with six groups reflecting their pre-hospital training level. We now examine the data for existing correlations between the phrase choice and the level of training. The data reveal that the participants who belong in the first three groups (Table 3.6) have similar preferences in terms of

template choice. In contrast, participants belonging to group 5 and 6 have distinctly different preferences[7]. We can therefore regroup the participants into 3 groups in terms of training level, summarizing groups 1-3 into one consistent preference group, we call the "novice" group, whereas we treat groups 5 and 6 as distinct groups with different levels of expertise.

Regarding the breathing rate, it is observed that the novice group (Groups 1-3), mostly prefer the verbose descriptions (template 2 and 4, 53%), followed by succinct (template 3 and 5, 20%) and finally the average (template 1, 16%), whereas Group 5 mostly prefer the succinct way (template 3 and 5, 77%), then the inference (template 6, 15%) and finally the verbose way (template 2 and 4, 57%). Medical doctors (Group 6) prefer both the verbose (56%) and succinct (40%) way. The average way was preferred by only 6% of the doctors. Similar observations are made for the Blood Oxygen Saturation parameter. Finally, for the heart rate parameter, it is observed that all user groups prefer the phrases that describe the trend either in a verbose or a succinct way (90% of Groups 1-3, 92% of Group 5 and 86% of Group 6). If we split these percentages, again the novice group prefer the verbose way over the succinct (60% and 30% respectively). Group 5 prefer the succinct way over the verbose (69% and 23% respectively) and finally 53% of the doctors prefer the verbose way and 33% the succinct.

These results suggest that the group preferences may vary, but there are common elements within their preferences. For instance, we found a general preference on reporting the trend of the physiological data across all groups (template 2 and 3), but a group-specific preference for the verbose over the succinct way. Doctors' preferences (Group 6) are not quite clear as to whether they prefer the verbose over the succinct way. We think that this might be due to the fact that doctors usually communicate their findings to groups with different expertise in a different manner, e.g. explaining results to patients or discuss a condition with other doctors, so they are inclined to customise their descriptions to the interlocutor.

[7]Note that none of our participants belong to Group 4, as mentioned in Section 3.4.3.

3.4.4.3 Gender / Phrase Choice Relation

There are no observed significant correlations between template choice and gender. When we examine the Breathing Rate parameter, 54% of male and 47% of female participants choose to refer to it a verbose way. On the other hand, 33% of females choose the succinct way, but only 17% of males (template 2 and 4). Whereas, for the Heart Rate parameter, we observe a different pattern: 86% of male participants choose the verbose way, but only half of the female participants (56%); 40% of women choose the succinct way, whereas 26% of men would choose the succinct way. We conclude that for generating data descriptions, we do not need to take user gender into account, as gender preferences vary for different parameters (confirming our findings in Section 4.1).

3.4.4.4 Experience with Sensor Data / Phrase Choice Relation

We finally investigate the influence of previous experience with sensor data has on template choice preference. We find that prior experience with sensor data yields mixed results. For the Breathing Rate parameter, there is no significant difference in preference between people with prior experience and people without. For the Blood Oxygen Saturation parameter, we observe that, on the one hand, the majority (58%) of the users without prior experience in sensor data prefer the verbose way of referring to the data trend, whereas only 30% of the users with prior experience prefer the verbose way. On the other hand, 42% of users with prior experience prefer the succinct way of referring to the data. 22% of the same group preferred referring to the average. And an almost equal percentage (21%) of the group without previous experience prefer the average reference. Finally, regarding the Heart Rate Parameter, users with prior experience almost equally prefer the verbose (both by referring to the trend and describing the value range) and the succinct way.

3.4.5 Discussion

The analysis highlights that grouping prospective users in terms of demographic characteristics is inappropriate. There is no clear evidence on what influences the participants' preferences. Therefore, in Chapter 6, we will use cluster analysis to group users in terms of preferences and then we utilise this information to address unknown users. We hypothesise that a new unknown user will belong to one of the existing user clusters. If the preferences of each cluster are modelled as optimisation function, we can apply multi-objective optimisation to find near optimally preferential content and thus satisfy unknown users.

3.5 Conclusions

In this chapter, we presented the overall NLG framework and the two domains that are employed in this thesis. The NLG framework consists of a data analysis module, a content selection module and template-based surface realiser. In the rest of the thesis, we will only be concerned with the content selection module. Two data collections in two different domains were presented: student feedback generation and health informatics. In Chapters 4 and 5, we will be using the data from the student feedback domain and Chapter 6 is concerned with the health domain. The feedback generation domain will be used for exploring data-driven methods for content selection and for addressing `known` users. The health domain introduces the challenge of handling `unknown` users and therefore it is used in Chapter 6.

Chapter 4

Comparison of Data-driven Approaches to Data-to-Text

This chapter explores data-driven methods for content selection of time-series data. It investigates whether the task of content selection can be formulated and solved equally effectively by using different data-driven techniques with respect to user preferences (RQ1). We consider the task of content selection for feedback summary generation for university students describing their performance during the lab of a Computer Science module over the semester as described in Chapter 3. *Factors* such as difficulty of the material, other deadlines etc. have two important qualities: (1) they change over time and (2) they can display some kind of relationship with each other. Specifically, the decision of factor selection can be influenced by:

- other factors that their values are correlated with (e.g. mentioning the health_issues when the marks decrease, see also Chapter 3),
- can be based on the selection of other factors to be included in the summary (e.g. minimising redundancy), and
- can be based on the factors' trends/behaviour over time.

For example, some factors may have to be discussed together in order to achieve some communicative goal. For instance, a teacher might want to refer to a student's marks as

a motivation for increasing the number of hours the student studied. Feedback generation can therefore be formulated in two ways: (1) as an incremental task, where content selection decisions are dependent on previous decisions, and (2) as a classification task where all variables/features are taken into account simultaneously in order to capture potential dependencies.

Data-driven methods can be more domain-independent and easier transferable to other domains than rule-based systems. In addition, they require less domain expert knowledge than rule-based approaches when datasets are available and can easily adapt to different users (Lemon, 2011). Recent work on report generation has started to move away from hand-written rules to data-driven techniques inspired by other areas of computational linguistics. The techniques include statistical techniques from Machine Translation (Belz and Kow, 2010) and supervised machine learning (Sowdaboina et al., 2014; Barzilay and Lapata, 2005). NLG systems have the potential to be more expressive and scalable through statistical approaches rather than rule-based approaches. For content selection, data-driven methods can automatically induce rules for text generation which can be close to the ones that experts use when summarising. Moreover, experts are hard to recruit and often experts provide dissimilar text, which makes it hard to translate text into rules. Regarding content selection, rules learnt from data corpora can be faster derived and thus cheaper. Due to their ability to take into account both large and small corpora, their coverage can be extended by using more training examples and from a few example can generalise for unseen scenarios. In contrast, acquiring knowledge from experts for constructing rules can be expensive, time-consuming and in some cases infeasible. In this thesis, the task of content selection is data-driven, whereas the task of surface realisation is handled through hand-crafted templates. This design is appropriate for studying content selection without introducing confounding variables from the surface text.

The contributions of the work presented in this chapter span across the following three sections:

Method(s) in Question	**Comparison**	**Results in Simulation**	**Results with students**
Reinforcement Learning (RL) (Section 4.1)	- Baseline 1: Rule-based system - Baseline 2: Brute Force system - Baseline 3: Lecturer-constructed summaries - Baseline 4: Random system	RL scores higher with respect to reward function than the other systems.	RL and Lecturer-constructed summaries are ranked similarly (no significant difference).
Multi-label Classification (MLC) (Section 4.2)	- Decision Trees without history - Decision Trees with predicted history - Majority Class baseline.	MLC performs significantly better in accuracy, precision, recall and F-score (Z-score, $p < 0.05$) than the baselines.	N/A
RL vs. MLC (Section 4.3)	- Baseline 1: Rule-based - Baseline 4: Random system	MLC performs best in accuracy, RL achieves highest reward.	Both are equally rated (no significant difference).

Table 4.1: Overview of the experiments of Chapter 4.

- Section 4.1 presents a novel and efficient method for tackling the challenge of content selection using a Reinforcement Learning (RL) approach. It also presents a preliminary evaluation in simulation and with students and discusses results. The content selection task is formulated as a Markov Decision Process and Reinforcement Learning is used for solving it, following previous work by Rieser and Lemon (2011). To our knowledge, this is the first effort of applying RL to a data-to-text application.

- Section 4.2 treats the content selection task as a supervised learning problem. Section 4.2.1 presents an innovative supervised learning approach to content selection using Multi-label classification (MLC). Section 4.2.2 presents a comparison of this method with a majority baseline and two binary classification methods: decision trees with history and without history.

- Section 4.3 presents a comparison of Multi-label classification with RL and discusses the strengths and the limitations of each approach.

Table 4.1 shows how the experiments are organised across the chapter.

Finally, Section 4.4 summarises the chapter. As mentioned previously, the focus of this thesis is on content selection. Surface realisation is performed through 29 handcrafted templates[8]. As described in the previous chapter 'template' is a quadruple consisting of an *id*, a *factor* (e.g. marks, understandability), a *reference type* (trend, weeks, average, other) and *surface text*.

4.1 Content Selection as a Reinforcement Learning Task

Reinforcement Learning (RL) is a machine learning technique that defines how an agent learns to take optimal actions so as to maximise a `cumulative` reward (Sutton and Barto, 1998). RL treats content selection as a `sequential` optimisation problem leading to optimal content selection. RL is inspired by the way humans learn to perform activities, such as walking. It is based on Skinner's theory of behaviourism (Skinner, 1938), which states that people tend to repeat actions that provide them with some reward (positive reinforcement) and avoid actions that "punish" them (negative reinforcement). Skinner proposes that people will perform an action that punishes them, if it eventually leads them to some greater final reward (expected reward).

A Markov Decision Process (MDP) is the mathematical formulation of a sequence of decisions, which is used for studying optimisation problems. For feedback generation, the decisions that affect content selection are made sequentially and at each time-stamp there are more than one actions available (content to be selected). As the goal here is to optimise the feedback summary, content selection can be seen as an MDP where the goal of the learning agent is to learn to take the sequence of actions that leads to optimal

[8]There are fewer than 36 templates, because for some factors there are less than 4 possible ways of referring to them.

content selection in an incremental manner. For instance, consider an agent that is designed to generate feedback for a given student. In the initial state, no feedback is written by the agent. Therefore, the agent might initially take an action and choose to refer to the marks achieved by the student. This action will introduce a state change, as the feedback will now have one sentence written. The agent is awarded some numerical *reinforcement signals (rewards)*, either positive or negative. Let's assume that the agent is being awarded a positive reward for mentioning the marks achieved. Next, the agent, after analysing the environment, might decide to comment on the hours the student studied over the semester. This action will again introduce a state change and the agent will be given some reward. However, the reward might be negative at this point, as this factor might not be useful for this particular student. For instance, this student might have scored high marks and have studied many hours per week, thus mentioning the hours the student studied adds no value to the feedback and makes the report longer than necessary.

For feedback generation, an MDP model is defined as a quadruple $<S, A, T, R>$, where:

- $S = \{s_0, s_1 ... s_N\}$, is a set of possible states that the agent can reach when taking actions. In our feedback generation task, the states consist of the time-series data related to a student's performance and the selected content (learning factors to talk about, e.g. marks). This encoding of the states helps to fulfil the *Markov Property*, `i.e.` the future states of the process depend only on that state and not on prior states (Markov, 1954). Therefore, each state needs to include information that will assist the agent in making decisions based on only the current state. For the feedback generation task, there are overall $N = 362,880$ possible states that the agent can reach. Each state consists of a description of the factor trends and the number of templates that have been selected so far. These features can be formalised into a state as follows:

```
( hours_studied: number (0-2),
understandability: number (0-2),
difficulty: number (0-2),
deadlines: number (0-2),
health_issues: number (0-2),
personal_issues: number (0-2),
lectures_attended: number (0-2),
revision: number (0-2),
marks: number (0-2),
length: integer,
selected_content: set of templates)
```

The number next to each factor denotes the trend of the factor (0 for increasing, 1 for decreasing, 2 for other). The integer next to length denotes the number of factors that have been selected for generation so far. The selected content includes the templates that have been selected for generation so far.

- $A = \{a_0, a_1 ... a_M\}$, is a set of possible actions that the agent can take. For feedback generation, the actions correspond to making decisions about referring to a learning factor or not. There are up to $M = 9$ possible actions available at each time corresponding to the 9 learning factors. Generally, the actions can be deemed as the means of exploration (attempting to discover new states by selecting sub-optimal actions) and exploitation (using the information in the state to achieve the best results that the agent is aware of). Every time that an action is selected to be mentioned, we use the following update rule for to inform the state which increments the length of the summary and introduces a state change:

Algorithm 1: The update rule.

if *decision is to mention a factor* **then**
  increment length;

- $R(s'|s,a)$, is a reward function that specifies the numerical reward that the agent receives after taking action a in state s resulting in state s'. The reward assists the agent in evaluating its actions and thus the decision making process. The Reward Function is discussed in detail in Section 4.1.1.

- $T : SxA \rightarrow S'$ is a probabilistic transition function and it provides a description of each action's effect in each state, i.e. performing an action a in state s will result in state s' with a probability $p(s'|s,a)$. For instance, choosing a particular factor to talk about, with what probability will the state change to s'? As mentioned previously, the time-series data can be characterised as `increasing`, `decreasing` or `other`. `Other` can characterise the marks of a student who scores the highest marks in all weeks apart from one, but can also characterise the marks of a student with variable performance. Therefore, it is uncertain whether the student has performed well or badly overall, as other context information is not included.

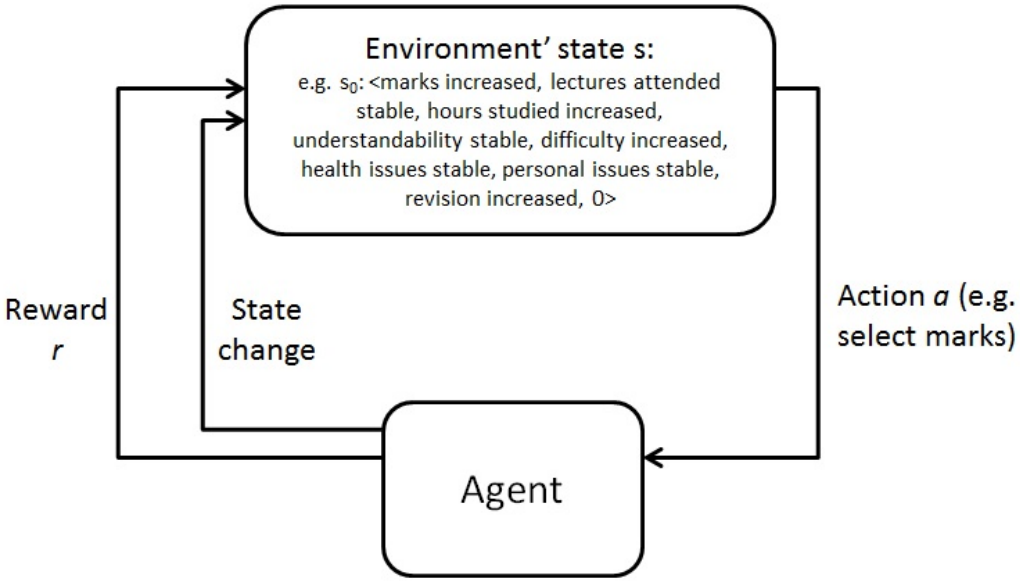


Figure 4.1: The RL setup.

Figure 4.1 shows how those modules are related. The dynamics of an MDP for content selection from time-series data can be described as follows. At the beginning of the interaction, at time stamp $t = 0$, the agent receives a representation of the

environment, which is the initial state s_0. It performs action a_0 which results in receiving reward r_0. This results in state s_1 at time step $t = 1$. This process can be seen as a finite sequence of states, actions and rewards $\{s_0, a_0, r_0, s_1, a_1, r_1, ..., s_t, a_t, r_t\}$. The Temporal-Difference (TD) learning method (Sutton and Barto, 1998) is used to train an agent for content selection. Any mapping from states to actions is called a policy π. The agent's goal is to learn an optimal policy denoted as π^*, a mapping from every state s to action a that will yield the highest expected return. Section 4.1.3 describes how the agent is trained.

Elements	n	Example
Factors	9	marks
States	362,880	see Section 4.1
Actions	9	mention health_issues
Templates	29	see Appendix A

Table 4.2: The RL elements.

4.1.1 Data-driven Reward Function

The reward function reflects the lecturers' preferences on summaries and is derived through linear regression analysis of a dataset containing lecturer constructed summaries and ratings of randomly generated summaries (as described in Section 3.3.4), following the *PARADISE* framework (Walker et al., 2000) and Rieser and Lemon (2011). The reward function models the content which significantly influences the lecturers' content selection decisions and ratings (0 - 100). Therefore, the reward function is fully informed by the data provided by the lecturers. We assumed that the optimal constructed summaries would score 100 (top rating available). We then transformed the summaries constructed by lecturers and the rated summaries into vectors in order to feed the linear regression model. An example of a vector is: $\{x_0, x_1, x_2, ..., reward\} = \{0, 1, 0, ..., 95\}$. The reward function is the following cumulative function:

$$Reward_{LECT} = intercept + \sum_{i=0}^{n} b_i * x_i + b_{90} * length \tag{4.1}$$

where x_i describes one combination of the data trends and a particular template. For instance, the value of x_1 is 1 if marks were increased and this trend is realised in the feedback, otherwise it is 0. In our domain, $n = 90$ in order to cover all the different combinations. The value of x_i is given by the function:

$$x_i = \begin{cases} 1, & \text{if the combination of a factor trend} \\ & \text{and a template type is included in a summary} \\ 0, & \text{if not.} \end{cases} \tag{4.2}$$

The *intercept* and the regression coefficients b have values that vary from -99 to 221. Finally, the *length* stands for the number of factors selected. A full description of X and the coefficients $\{B = b_0, b_1...b_n\}$ are provided in Appendix C. The reward is not only based on the selected content, but also on the way it is referred to, e.g. choosing the template that mentions the average or referring to the trend etc.

The *discounted (cumulative) reward* u_t denotes the expected reward that was collected in state s during the execution of action a and can be computed by:

$$u_t = r_{t+1} + \gamma^2 r_{t+2} + ... + \gamma^{t-1} r_T = \sum_{i=1}^{T} \gamma^{t-1} r_{t+1} \tag{4.3}$$

for a cycle that lasts for steps T. γ is a discount factor and is equal to 0.1.

The reward function is maximized (Reward = 861.85) for the scenario shown in Table 4.3 (please note that this scenario was not observed in the data collection). This means that if the column TREND described the data of a given student and if the templates described in column TEMPLATES where included in the feedback summary, then the constructed summary would have score the highest possible reward. However, this does not mean that a summary that scores less than the highest reward is not good enough. Due to the nature of the reward function, the agent can be trained for unseen scenarios such as the the one presented in Table 4.3. The reward function is minimized (Reward = -586.0359) for the scenario shown in Table 4.4 (please note that this scenario was not

observed in the data collection). Again, if a summary scores higher than the minimum reward, it does not mean that a summary is good enough. Each summary can only be compared to summaries constructed for the same scenario.

Factor	Trend	Template
difficulty	stable	NOT_MENTIONED
hours studied	stable	TREND
understandability	stable	NOT_MENTIONED
deadlines	increase	WEEKS
health issues	stable	WEEKS
personal issues	stable	WEEKS
lectures attended	stable	WEEKS
revision	stable	OTHER
marks	increase	TREND

Table 4.3: The scenario at which the reward function is maximised.

Factor	Trend	Template
difficulty	increase	AVERAGE
hours studied	stable	NOT_MENTIONED
understandability	decrease	AVERAGE
deadlines	*	*
health issues	increase	TREND
personal issues	stable	TREND
lectures attended	stable	NOT_MENTIONED
revision	stable	AVERAGE
marks	stable	TREND

Table 4.4: The scenario at which the reward function is minimised (* denotes multiple options result in the same minimum reward).

4.1.2 Temporal-Difference Learning

`Temporal difference (TD)` learning is a prediction method used for solving reinforcement learning tasks. In TD learning, the agent adjusts its current actions in a manner where its future actions will yield higher reward. An agent can therefore choose an action that yields less immediate reward, if this action will lead to higher long-term reward. In the case of feedback generation this means that the agent needs to make a decision on whether to refer to a factor or not. The algorithm can estimate how well this factor will be combined with other factors and whether the decision will lead to

high future reward, i.e. optimised content. This quality makes this learning approach appealing for our task, as our goal is to optimise the generated feedback summaries by predicting "how preferable the generated summary is".

We use a special case of TD-learning called Q-learning. In this case, the learnt action-value function directly approximates the optimal action-value function, independent of the policy being followed. This dramatically simplifies the analysis of the algorithm and enables early convergence proofs (Sutton and Barto, 1998). The policy still has an effect in that it determines which state-action pairs are visited and updated. However, all that is required for correct convergence is that all pairs continue to be updated. This is a minimal requirement in the sense that any method guaranteed to find optimal behaviour in the general case must require it. The Q-learning algorithm is shown in procedural below:

Algorithm 2: Q-learning algorithm.

```
Initialize Q(s,a) arbitrarily;
while s is not terminal do
    Initialize s;
    for each step of cycle do
        Choose a from s using ϵ - greedy policy;
        Take action a, observe r, s';
        Q(s,a) <- Q(s,a) + a[r +γ max'_a Q(s', a') - Q(s,a)];
        s <- s';
    end
end
```

4.1.3 Training

A time-series generation policy is trained with 9,250 runs using the algorithm described above (Sutton and Barto, 1998). After 8,750 runs the algorithm converged to the maximum reward value and therefore stopped. During the training phase, the learning agent generates feedback summaries. When the construction of the summary begins, the length of the summary is 0. The agent chooses a factor using an ϵ - greedy policy and decides whether to talk about it or not. If the agent decides to talk about the factor, the length of the summary increments and thus a state change is introduced.

It repeats the process until it decides for all factors whether to talk about them or not. The agent is finally rewarded at the end of the process using the Reward function described in Equation 4.1. The training stops when the stopping criterion is met. The stopping criterion defines that the training will stop when it reaches a maximum (= same reward) for 500 cycles (converges). Figure 4.2 shows the learning curve of the agent.

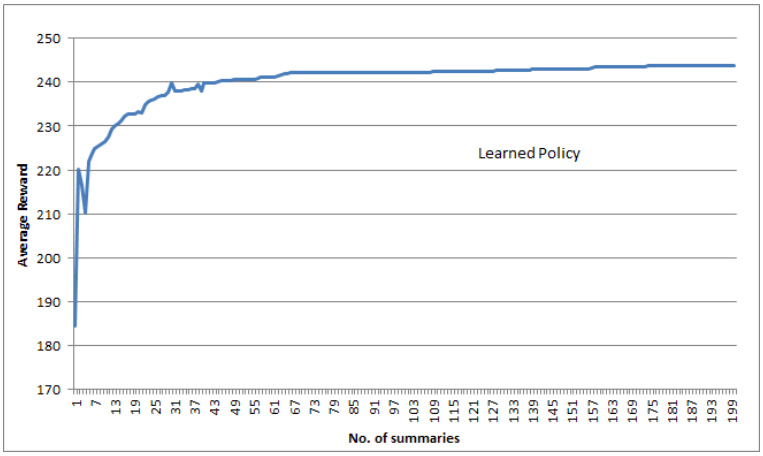


Figure 4.2: Learning curve for the learning agent. The x-axis shows the number of summaries produced and y-axis the total reward received for each summary. The number of summaries is averaged over 50 summaries.

4.1.4 Ordering

The `ordering` of content is kept fixed throughout the thesis, as we wanted to reduce the confounding variables. This allows us to study content selection in isolation, as content could otherwise be rated differently according to the ordering. Other NLG systems reported in the literature have kept the ordering of reports stable, such as the `STOP` system (Reiter et al., 1999).

In order to define the order in which the lecturers describe the factors, we use a corpus analysis method, following previous work as for instance (McKeown, 1985; Yu et al., 2007). The feedback summaries are transformed into n-grams of factors. The constructed n-grams are used to compute the bi-gram frequency of the tokens in order to identify which factor is most probable to be referred to initially, which factors follow

particular factors and which factor is usually talked about in the end. It was found that the most frequent ordering is: start, marks, hours_studied, understandability, difficulty, deadlines, health_issues, personal_issues, lectures_attended, revision, end.

4.1.5 Preliminary Evaluation

The RL system is evaluated in two ways. First, by using the reward function as a metric and second, by asking students to rate the generated summaries. In both evaluations, we compare feedback reports generated using the Reinforcement Learning agent with four other baseline systems described below:

Baseline 1: Rule-based system. This system selects factors and templates for generation using a set of rules. The hand-crafted rules are derived from a combination of the L&T expert's advice (by thinking aloud) and a student's preferences and is therefore a challenging baseline. The rules derived by this process are given in Appendix D.

Baseline 2: Brute Force system. This system performs a search of the state space, by exploring randomly as many different feedback summaries as possible. The Brute Force algorithm is shown below:

Algorithm 3: Brute Force algorithm.

```
Data: D
Result: best feedback
for n=0...10,000 do
    construct randomly feedback[n];
    assign reward[n] to feedback[n];
    if reward[n] > reward[n-1] then
        best feedback = feedback[n];
    else
        best feedback = feedback[n-1];
    end
end
```

In each run, the algorithm constructs a feedback summary, then it calculates its reward, using the same reward function used for the Reinforcement Learning approach described in Section 4.1.1, and if the reward of the new feedback is better than the previous, it keeps the new one as the best. It repeats this process for 10,000 times

for each scenario. Finally, the algorithm returns the summary that scored the highest ranking. Although Brute Force is an exhaustive search algorithm, here, instead of letting it explore all 362,880 possible states, we only allow it to run for 10,000, similar to the RL algorithm, in order to be able to directly compare them. Because its complexity is $O(n!)$, where n is the number of available content features, for tasks with a small feature vector, this algorithm is fast and efficient. However, for large state spaces this algorithm can be inefficient as the state space grows in a factorial manner.

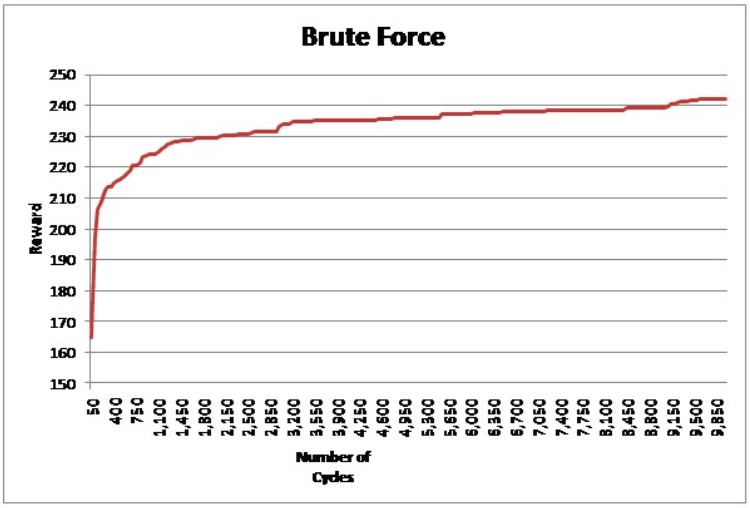


Figure 4.3: The graph shows the number of cycles that the Brute Force algorithm needs to achieve specific rewards.

Baseline 3: Lecturer-constructed summaries. These are the summaries produced by the lecturers, as described in Section 3.3.4, for Task 2 using template-generated utterances.

Baseline 4: Random system: This system constructs feedback summaries by selecting factors and templates randomly as described in Task 3 (in Section 3.3.4).

4.1.6 Results in Simulation

Table 4.5 presents the results of the evaluation performed using the data-driven Reward Function, comparing the RL policy with the four baseline systems. Each system generated 26 feedback summaries from the 26 scenarios. The RL policy scores significantly higher than any other baseline for the given scenarios ($p < 0.05$, paired t-test).

Time-Series Summarisation Systems	Reward
RL	**243.82**
Baseline 1: Rule-based	107.77
Baseline 2: Brute Force	**241.98**
Baseline 3: Lecturers	124.62
Baseline 4: Random	43.29

Table 4.5: The average rewards that are assigned to summaries produced from the different systems (bold signifies higher reward).

4.1.7 Evaluation with Students and Results

A subjective evaluation is conducted using 1st year students of Computer Science as participants. We recruited 17 students, who are all native-English speakers. The number of recruited students is lower than in the subsequent experiments, as these students were recruited through a general e-mail to the university's student mailing list. For the next experiments, we improved the recruitment process, by asking students to perform the evaluations while in the computer lab, so as to make sure that more students will complete the evaluations. The participants were shown 4 feedback summaries in a random order, one generated by the RL policy, one from the rule-based system (Baseline 1), one from the Brute Force system (Baseline 2) and one summary produced by a lecturer using the templates (Baseline 3). Given the poor performance of the Random system in terms of reward, Baseline 4 was omitted from this study. The participants had to rank the summaries in order of preference: 1 for the most preferred and 4 for the least preferred. Each participant repeated the process for four scenarios and the participant is allowed to opt out at any stage. A task-based evaluation would not be feasible for two reasons: (1) Due to ethical restrictions, we could not show to different subsets of students feedback summaries generated by different systems, because some students would be provided with less effective feedback summaries and this might impact on their performance, and (2) Due to time restrictions: Providing all students with the summary of each system each semester would require four semesters to complete the evaluation. To overcome this barrier, we asked students for their preferred feedback

summary and we used lecturers' ratings as means of quality assurance.

Overall there are 26 different scenarios, as described in Section 3.3.1. All summaries presented to a participant are generated from the same scenario. The mode values of the rankings of the preferences of the students are shown in Table 4.6. The web-based system used for the evaluation is shown in Figure 4.4 - page 88.

System	Mode of Rankings
RL	3rd
Baseline 1: Rule-based	**1st***
Baseline 2: Brute Force	4th
Baseline 3: Lecturer-constructed	3rd

Table 4.6: The mode value of the rankings of the preference of the students, * denotes significance at $p<0.05$, Mann-Whitney U and a Wilcoxon signed-rank test.

We ran a Mann-Whitney U and a Wilcoxon signed-rank test to evaluate the difference in the responses of our 4-point Ranking scale question between the RL system and the other three baselines. It is found that, for the given data, the students rank the feedback generated by the RL system similarly to the feedback produced by the experts, i.e. there is no significant difference between the mean value of the rankings of the RL system and the lecturer-produced summaries ($p = 0.8$, Mann-Whitney U and Wilcoxon test).

The preference of the users for the Brute Force system does not differ significantly from the summaries generated by the RL system ($p = 0.1335$, Mann-Whitney U and Wilcoxon test). However, the computational cost of the Brute Force is higher because each time that the algorithm sees a new scenario it has to run approximately 3k times to reach a good summary of 230 reward (as seen in Figure 4.3) and about 10k to reach the optimal one of 240 reward. In contrast, the RL agent inherently accounts for unseen scenarios.

Finally, the users significantly prefer the summaries produced by the Rule-based system (Baseline 1) to the summaries produced by the RL system ($p=0.015$). This is possibly due to the fact that in the rule-based system some knowledge of the end user's

preferences (i.e. students) is taken into account in the rules which is different from the other three systems. This fact suggests that students' preferences should be taken into account as they are the receivers of the feedback. This can also be generalised to other areas, where the experts and the end users are not the same group of people. As the RL policy is not trained to optimise for the evaluation criteria, in the next chapter, we will explore reward functions that bear in mind both the expert knowledge and the students' preferences. Finally, as mentioned in the beginning of the chapter, rule-based systems can be expensive and difficult to build due to the fact that they require domain expert knowledge.

In conclusion, a statistical learning approach to summarisation from time-series data in the area of feedback reports is presented. We show a way of constructing a data-driven reward function from lecturer constructed summaries that can capture dependencies between the time-series data and the realisation phrases. Finally, the preliminary evaluation shows that students rank the RL generated reports similarly to the lecturer-constructed ones, although the output is quite different. Generally, for language generation there is no right or wrong answer; preferences on text are quite subjective. In addition, it is evident from our results that the students preferred the rule-based system to all other systems. We suspect that this is due to the quality of the dataset used for training. Lecturers gave feedback summaries that were quite variable, and there was no mechanism to assess the quality. As data-driven approaches are influenced by the dataset quality, learnt systems might not produce as good summaries as expected. This could also apply to rule-based systems, however it seems that our rule-based system is of high quality.

In the next section, we will explore a supervised learning approach, which is able to generate summaries similar to lecturer-constructed. Finally, we will discuss how students and lecturers rate summaries constructed by the RL and the supervised system in a final evaluation.

Raw Data

factors	week 2	week 3	...	week 10
marks	5	4	...	5
hours_studied	1	2	...	3
...	...	...	...	...

Trends from Data

factors	trend
(1) marks (M)	trend_other
(2) hours_studied (HS)	trend_increasing
(3) understandability (Und)	trend_decreasing
(4) difficulty (Diff)	trend_decreasing
(5) deadlines (DL)	trend_increasing
(6) health_issues (HI)	trend_other
(7) personal_issues (PI)	trend_decreasing
(8) lectures_attended (LA)	trend_other
(9) revision (R)	trend_decreasing

Summary

Your overall performance **was excellent** during the semester. Keep up the good work and maybe try some more challenging exercises. Your attendance was **varying** over the semester. Have a think about how to use time in lectures to improve your understanding of the material. You spent **2 hours studying the lecture material on average**. You should dedicate more time to study. You seem to find the material **easier to understand compared to the beginning of the semester**. Keep up the good work! You revised **part of** the learning material. Have a think whether revising has improved your performance.

Table 4.7: The table on the top left shows an example of the time-series raw data for feedback generation. The table on the bottom left shows an example of described trends. The box on the right presents a target summary (target summaries have been constructed by teaching staff).

4.2 Content Selection as a Supervised Task

The content selection task for feedback generation can be formulated as a classification task as follows: given a set of 9 time-series factors, select the content that is most appropriate to be included in a summary. Content is regarded as labels (each template represents a label) and thus the task can be thought of as a classification problem. As mentioned in Chapter 3, there are 4 ways to refer to a factor: (1) describing the trend, (2) describing what happened at every time stamp, (3) mentioning the average or (4) making another general statement. Overall, for all factors there are 29 different templates (Appendix A). An example of the input data is shown in Table 4.7. There are two decisions that need to be made: (1) whether to talk about a factor and (2) in which way to refer to it. Instead of dealing with this task in a hierarchical way, where

the algorithm will first decide whether to talk about a factor and then to decide how to refer to it, we formulate the task in a way that reduces the learning steps. Therefore, classification can reduce the decision workload by deciding either in which way to talk about it, or not to talk about a factor at all.

For the *binary classification*, the classifier learns to predict whether a template is to be included in a summary or not. Essentially, one needs as many classifiers as the templates, as for each template the decision to be included in a summary or not is made independently (see also Figure 4.5). In *Multi-label classification*, the classifier learns to predict **a set of labels** that correspond to each instance, where each template corresponds to a label. As each template contains information about the factor (marks etc.) and the way to refer to it (trend, average etc.) the complexity is reduced. Content selection as a classification task is not a new challenge. Collective content selection (Barzilay and Lapata, 2005) is similar to our proposed method in that it is a classification task that predicts multiple templates from the same instance simultaneously. The difference between the two methods lies in that the collective content selection requires the consideration of an individual preference score, which is defined as the preference of the entity to be selected or omitted. The preference score is based on the values of entity attributes and is computed (1) using a boosting algorithm, which is based on ensemble of algorithms and (2) the identification of links between the entities with similar labels. In contrast, ML classification does not need the computation of links between the data and the templates. ML classification can be also applied to other tasks where features are correlated, such as text classification, movie genre categorisation etc. (Tsoumakas et al., 2010).

4.2.1 Multi-label Classification

Classification is concerned with the identification of a category l from a set of disjoint categories L (with $|L| > 1$) that an instance belongs to, given the characteristics of the instance. If $|L| = 2$, then the learning task is called `binary classification`; for

example a task where a classifier is trained to associate e-mails with either spam or not (i.e. 1 or 0, and hence binary). If $|L| > 2$, then the learning task is called `multi-class classification`; for example a task where the classifier can associate a running area as good, bad or ok. In `Multi-label` classification (MLC), the instances are associated with a set of labels $Y \subseteq L$ (Tsoumakas et al., 2010). For example, a newspaper article can be classified into `health`, `science`, `economy`, `politics`, `culture` etc. A specific news article concerning the breakthrough of the Ebola cure can be classified into both of the categories `health` and `science`. In the same way, students' data can be assigned labels that describe them. Each label corresponds to a template. The set of chosen templates can then form a feedback summary.

In Chapter 3, we showed that the learning factors are correlated with each other. Multi-label classification is efficient in taking data dependencies into account and generating a set of labels, in our case templates, simultaneously (Tsoumakas et al., 2010). In addition, we observe that different lecturers tend to choose different templates when constructing feedback for the same student. Therefore, in our dataset, one set of factor values can result in various sets of templates as interpreted by the different experts. An ML classifier is able to make decisions for all templates simultaneously and capture these differences.

MLC algorithms have been divided into three categories: algorithm adaptation methods, problem transformation and ensemble methods (Tsoumakas and Katakis, 2007; Madjarov et al., 2012). Algorithm adaptation approaches extend simple classification methods to handle multi- label (ML) data. For example, the k-nearest neighbour algorithm is extended to ML-kNN by (Zhang and Zhou, 2007). ML-kNN identifies for each new instance its k nearest neighbours in the training set and then it predicts the label set by utilising the maximum a posteriori principle according to statistical information derived from the label sets of the k neighbours. Problem transformation approaches transform the MLC task into one or more simple classification tasks. Ensemble methods are algorithms that use ensembles to perform ML learning and they are

based on problem transformation or algorithm adaptation methods. In this thesis, we applied RAkEL (Random k-labelsets) (Tsoumakas et al., 2010): an ensemble problem transformation method, which constructs an ensemble of single-label classifiers, where each one deals with a random subset of the labels.

RAkEL is based on Label Powerset (LP), a problem transformation method. Label Powerset treats every labelset as a single-class label in a multi-class task. LP benefits from taking into consideration label correlations, but does not perform well when trained with few examples as in our case (Tsoumakas et al., 2010). For instance, our dataset could include up to 2^9 (=$536,870,912$) distinct classes if it was treated as a multi-class problem, although in a real case they would be much fewer). In addition, our dataset consists only of 39 instances. RAkEL overcomes this limitation by constructing a set of LP classifiers, and each classifier is trained with different random subsets of the set of labels (Tsoumakas et al., 2010). In the end, it uses a majority voting scheme to make predictions.

The LP method transforms the ML task into one single-label multi-class classification task, where the possible set of predicted variables for the transformed class is the powerset of labels present in the original dataset. For instance, the set of labels L = $\{temp_0, temp_1, ...temp_{28}\}$ could be transformed to the powerset of all possible combinations LP =$\{temp_{0,1,2}, temp_{28,3,17,}...\}$. This algorithm does not perform well when considering a large number of labels, as the label space grows exponentially which results in overfitting (modelling error instead of the relationship). RAkEL tackles this problem by constructing an ensemble of LP classifiers and training each one on a different random subset of the set of labels (Tsoumakas et al., 2010). The algorithm was implemented using the MULAN Open Source Java library (Tsoumakas et al., 2011), which is based on WEKA (Witten and Frank, 2005). The algorithm works in two phases:

1. the production of an ensemble of LP algorithms, and
2. the combination of the LP algorithms.

4.2.1.1 The Production Phase of RAkEL

RAkEL takes as input the following parameters: (1) the number of iterations m, which is developer-specified and denotes the number of models that the algorithm will produce as it constructs one model per classifier, (2) the size of labelset k, which is also developer-specified, (3) the set of labels L, and (4) the training set D. During the initial phase it outputs an ensemble of LP classifiers and the corresponding k-labelsets. A pseudocode for the production phase is shown below:

Algorithm 4: RAkEL production phase.

```
Input  : iterations m, k labelsets, labels L, training data D
Output: the ensemble of LPs with corresponding k-labelsets
for  i=0... m do
    Select random k-labelset from L;
    Train an LP on D;
    Add LP to ensemble;
end
```

4.2.1.2 The Combination Phase

During the combination phase, the algorithm takes as input the results of the production phase, i.e. the ensemble of LPs with the corresponding k-labelsets, the set of labels L, and the new instance x and it outputs the result vector of predicted labels for instance x. During run time, RAkEL estimates the average decision for each label in L and if the average is greater than a threshold t (determined by the developer), it includes the label in the predicted labelset. We use the standard parameter values of t, k and m (t = 0.5, k = 3 and m = 58 (2*29 templates)), which are empirically found best.

4.2.2 Binary Classification - Decision Trees

Content selection can be framed as a binary classification task: given a set of time-series data, decide for each template separately whether it should be included in a summary or not. In the field of multi-label classification this approach is known as `Binary Relevance` (Tsoumakas and Katakis, 2007). Binary classification assumes that the

templates are independent of each other, thus the decision for each template is taken in isolation from the others, which is not appropriate for our domain. In order to capture the dependencies in the context or history, multiple simple classifiers can make the decisions for each template iteratively. After each iteration, the feature space grows by 1 feature, in order to include the history of the previous template decisions. In the Multi-label Classification field, this approach is called `Classifier Chains` (Tsoumakas and Katakis, 2007). We apply several different supervised learning algorithms including JRip, Decision Trees (C.4.5), Naive Bayes, k-nearest neighbour, logistic regression, multi-layer perceptron and Support Vector Machines using WEKA. It is found that Decision Trees achieve on average 3% higher accuracy than all other algorithms ($p < 0.05$, Z-test). We therefore went on to use Decision Trees that use generation history in three ways.

Firstly, for **Decision Tree (no history - see also Figure 4.5)**, 29 decision-tree classifiers were trained, one for each template. The input of these classifiers were the 9 factors and each classifier was trained in order to decide whether to include a specific template or not. This method did not take into account other selected-templates - it was only based on the time series data. An example of the rules derived from such a decision tree can be shown in Appendix E (Algorithm 6).

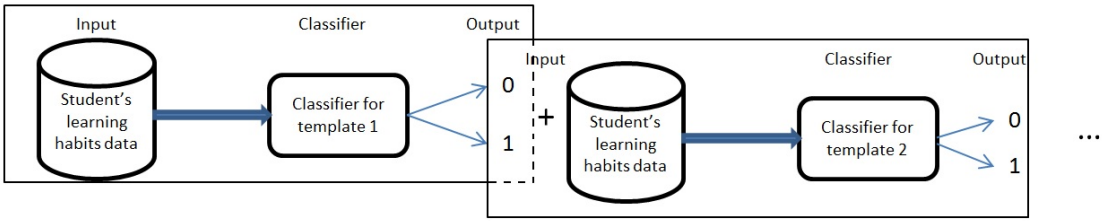


Figure 4.6: Feedback generation as a binary classification problem with history. 29 classifiers need to be trained, each one is responsible for each template. This time the input consists not only of the student's learning habits but also the previous decisions made on previous templates.

Secondly, for **Decision Tree (with predicted history - see also Figure 4.6)**, 29 classifiers were also trained, but this time the input included the previous decisions made by the previous classifiers (i.e. the history) as well as the set of time-series data in

Classifier	**Accuracy** (10-fold)	**Precision**	**Recall**	**F-score**
Decision Tree (no history)	*75.95%	67.56	75.96	67.87
Decision Tree (with predicted history)	**73.43%	65.49	72.05	70.95
Majority-class (single label)	**72.02%	61.73	77.37	68.21
MLC - RAkEL (no history)	**76.95%**	**85.08**	**85.94**	**85.50**
Decision Tree (with real history)	****78.09%**	74.51	78.11	75.54

Table 4.8: Average, precision, recall and F-score of the different classification methods (t-test, * denotes significance with $p < 0.05$ and ** significance with $p < 0.01$, when comparing each result to RAkEL).

order to emulate the dependencies in the dataset. For instance, classifier n was trained using the data from the 9 factors and the template decisions for templates 0 to $n-1$. An example of the rules derived from such a decision tree can be shown in Appendix E (Algorithm 7).

When history is included, the trees display more complex structure, and a greater number of rules is derived. This structure allows for capturing dependencies between factors and templates and it is more detailed. Therefore, we expect that including history will contribute to generating more accurate summaries.

Thirdly, for **Decision Tree (with real history)**, the real, expert values were used rather than the predicted ones. The above-mentioned classifiers are compared with the **Majority-class (single label)** baseline, which labels each instance with the most frequent template. The next session presents a comparison of MLC-RAkEL (no history) with the aforementioned iterated classification approaches (Section 4.2.3).

4.2.3 Comparison of Multi-label Classification with Binary Classification

The accuracy, the weighted precision, the weighted recall, and the weighted F-score of the classifiers are shown in Table 4.8. `Accuracy` is calculated as the proportion of the correctly classified templates to the population of classified templates of the test set (Mitchell, 1997). `Precision` denotes the fraction of the generated templates that

are relevant. It is calculated as the proportion of relevant generated templates to the set of generated templates. `Recall` denotes the fraction of relevant templates that are generated. It is estimated as the proportion of relevant generated templates to the set of relevant templates. Finally, the `F-score` or `F-measure` is the harmonic mean of precision and recall and is estimated by the following equation:

$$F = 2 * (precision * recall)/(precision + recall) \tag{4.4}$$

It is found that in 10-fold cross validation RAkEL performs significantly better in all these automatic measures (accuracy = 76.95%, F-score = 85.50% , t-test, p <0.05). Remarkably, MLC-RAkEL (no history) achieves more than 10% higher F-score than the other methods (Table 4.8). The average accuracy of the single-label classifiers is 75.95% (10-fold validation), compared to 73.43% of classification with history. The reduced accuracy of the classification with predicted history is due to the error in the predicted values. In this method, at every step, the predicted outcome is used including the potentially incorrect decisions that the classifier made. The upper-bound accuracy is 78.09% calculated by using the experts' previous decisions and not the potentially erroneous predicted decisions. This result is indicative of the significance of the relations between the factors showing that the predicted decisions are dependent due to existing correlations as discussed in Chapter 3, therefore the system should not take these decisions independently. MLC-RAkEL (no history) performs better due to its capability to take into account the relationships and dependencies in the data.

4.3 Comparison of Supervised Learning with Reinforcement Learning

As we show in the previous section, RAkEL performs better than any other classification algorithm in terms of accuracy, precision, recall and F-score. Therefore, we compare the MLC-RAkEL (no history) system and the RL system with two baselines by measuring

Time-Series Summarisation Systems	**Accuracy**	**Reward**	**Rating Mode (mean)**	**Data Source**
MLC-RAkEL (no history)	**85%**	65.4	7 (6.24)	Lecturers' constructed summaries
Reinforcement Learning	**66%	**243.82**	**8 (6.54)**	Lecturers' ratings & summaries
Rule-based	**65%	107.77	7, 8 (5.86)	L&T expert
Random	**45.2%	43.29	*2 (*4.37)	Random

Table 4.9: Accuracy, average rewards (based on lecturers' preferences) and averages of the means of the student ratings. Accuracy significance (Z-test) with MLC-RAkEL (no history) at $p < 0.05$ is indicated as * and at $p < 0.01$ as **. Student ratings significance (Mann Whitney U test) with MLC-RAkEL (no history) at $p < 0.05$ is indicated as *.

the accuracy of their outputs, the reward achieved by the reward function used for the RL system, and finally we perform evaluation with students. Each of the four systems (two baselines, the RL and the ML system) generated 26 feedback summaries corresponding to the 26 student profiles. These summaries are evaluated in simulation and with real student users. In order to reduce the confounding variables, we kept the ordering of content in all systems the same, by adopting the ordering of the rule-based system, as described in Section 4.1.5. The two baselines are as follows:

1. **Rule-based System:** as described in Section 4.1.5 - page 70.

2. **Random System:** initially it selects a factor randomly and then selects a template randomly, until it makes decisions for all factors - page 67.

4.3.1 Results in Simulation

Table 4.9 presents the accuracy and reward in simulation of each algorithm when used to generate the 26 summaries. *Accuracy* measures how similar the generated output (test set) is to the gold standard (training set), whereas the reward function calculates a score regarding how good the output is, given an objective function. In order to have an objective view on the results, the score achieved by each algorithm using the reward function is also calculated. MLC-RAkEL (no history) achieves significantly higher

accuracy, which is expected as it is a supervised learning method and learns from the given training data. The rule-based system and the RL system have lower accuracy compared to the MLC-RAkEL (no history) system. There is evidently a mismatch between the rules and the test-set; the content selection rules are based on heuristics provided by an L&T Expert rather than by the same pool of lecturers that created the test-set. The RL is trained to optimise the selected content and not to replicate the existing lecturer summaries, hence there is a difference in accuracy.

RL is trained to optimise for this function, and therefore it achieves higher reward, whereas MLC-RAkEL (no history) is trained to learn by examples, therefore it produces output closer to the gold standard (lecturer's produced summaries). RL uses exploration and exploitation to discover combinations of content that results in higher reward. The reward represents predicted ratings that lecturers would give to the summary. The reward for the lecturer produced summaries is 124.62 and for the MLC method is 107.77. The MLC system performs worse than this gold standard in terms of reward, which is expected given the error in predictions (supervised methods learn to reproduce the gold standard and they are not trained to optimise for an objective function). Moreover, each decision is rewarded with a different value as some combinations of factors and templates have greater or negative regression coefficients. For instance, the combination of the factors "deadlines" and the template that corresponds to <weeks> is rewarded with 57. On the other hand, when mentioning the <average> difficulty, the summary is "punished" with -81 (see also description of the reward function in Section 4.1). Consequently, a single poor decision in the MLC can result in much less reward.

4.3.2 Subjective Results with Students

37 first year computer science students participated in the study. Each participant is shown a graphical representation of the time-series data of one student and four different summaries generated by the four systems (see Figure 4.7 - page 90). The order of the

presented summaries is randomised. They are asked to rate each feedback summary on a 10-point Rating scale in response to the following statement: "Imagine you are the following student. How would you evaluate the following feedback summaries from 1 to 10?", where 10 corresponds to the most preferred summary and 1 to the least preferred. Earlier in this chapter, we presented a reward function based on lecturers' preferences. In the next chapter, we will explore a reward function that is based on students' preferences. Therefore, we asked students to rate the summaries instead of ranking them. We will discuss how we utilised these ratings in the next chapter.

The difference in ratings between the MLC-RAkEL (no history) system, the RL system and the Rule-based system is not significant (see Table 4.9, Mann-Whitney U test, $p > 0.05$). However, there is a trend towards the RL system ($p = 0.06$) when compared to the MLC-RAkEL (no history) system. The classification method reduces the generation steps, by making the decisions of the factor selection and the template selection jointly. Finally, the students significantly prefer all the systems over the random system.

4.4 Conclusions

In this chapter, we presented two main approaches to content selection of time-series data in the context of feedback generation for university students. The first treats content selection as a sequential task using RL. The second considers content selection simultaneously using multi-label classification. Those approaches were compared against four baseline systems: Rule-based system, Brute Force system, Lecturer-constructed summaries and Random system. The results obtained for the evaluation with students and from automatic metrics suggest that:

1. Reinforcement Learning is able to achieve optimal solutions in fewer cycles than a Brute Force (exhaustive search) algorithm, due to its capability to explore the search space and exploit the already obtained results.

2. Multi-label classification is more efficient and performs better in terms of accuracy, precision, recall and F-score compared to other supervised approaches. However, the quality of the output depends on the quality of the training set.

3. Both Multi-label classification and Reinforcement Learning approaches perform comparably when rated by students. In this chapter, we evaluated our systems with students, as students are the receivers of feedback.

We have shown that MLC-RAkEL (no history) for summarisation of time-series data has an accuracy of 76.95% and that this approach significantly outperforms other classification methods as it is able to capture dependencies in the data when making content selection decisions. MLC-RAkEL (no history) is also directly compared to a RL method. It is found that although MLC-RAkEL (no history) is almost 20% more accurate than RL. However, both methods perform comparably when rated by humans. This may be due to the fact that the RL optimisation method is able to provide more varied responses over time rather than just emulating the training data as with standard supervised learning approaches. Foster (2008) demonstrated similar results when performing a study on generation of emphatic facial displays. In our study, the human ratings correlate well to the average scores achieved by the reward function. However, the human ratings do not correlate to the accuracy scores. It is interesting that the two methods that score differently on various automatic metrics are evaluated similarly by users. Another issue that typically arises from supervised learning is its limited ability to generalise to unseen scenarios.

The comparison shows that each method can serve different goals. Multi-label classification generates output closer to gold standard whereas RL can optimise the output according to a reward function, typically requiring less training data than in supervised settings. Another advantage of RL is that it can generalise well to unseen scenarios. MLC could be used when the goal of the generation is to replicate phenomena seen in the dataset, because it achieves high accuracy, precision and recall. However, optimisation methods can be more flexible, provide more varied output and can be

trained for different goals, e.g. for capturing preferences of different users.

Finally, it is observed that students and lecturers have different preferences on what constitutes a useful feedback summary. The students rated the summaries produced by the Rule-based system, which takes into account students' preferences, considerably higher than the summaries generated by the other systems. In the next chapter, we will discuss students' and lecturers' preferences and will develop a system that adapts to all stakeholders. Therefore, we introduce a new task which aims to consider students' and lecturers' preferences simultaneously when generating feedback. We will call this task *Multi-adaptive Natural Language Generation*.

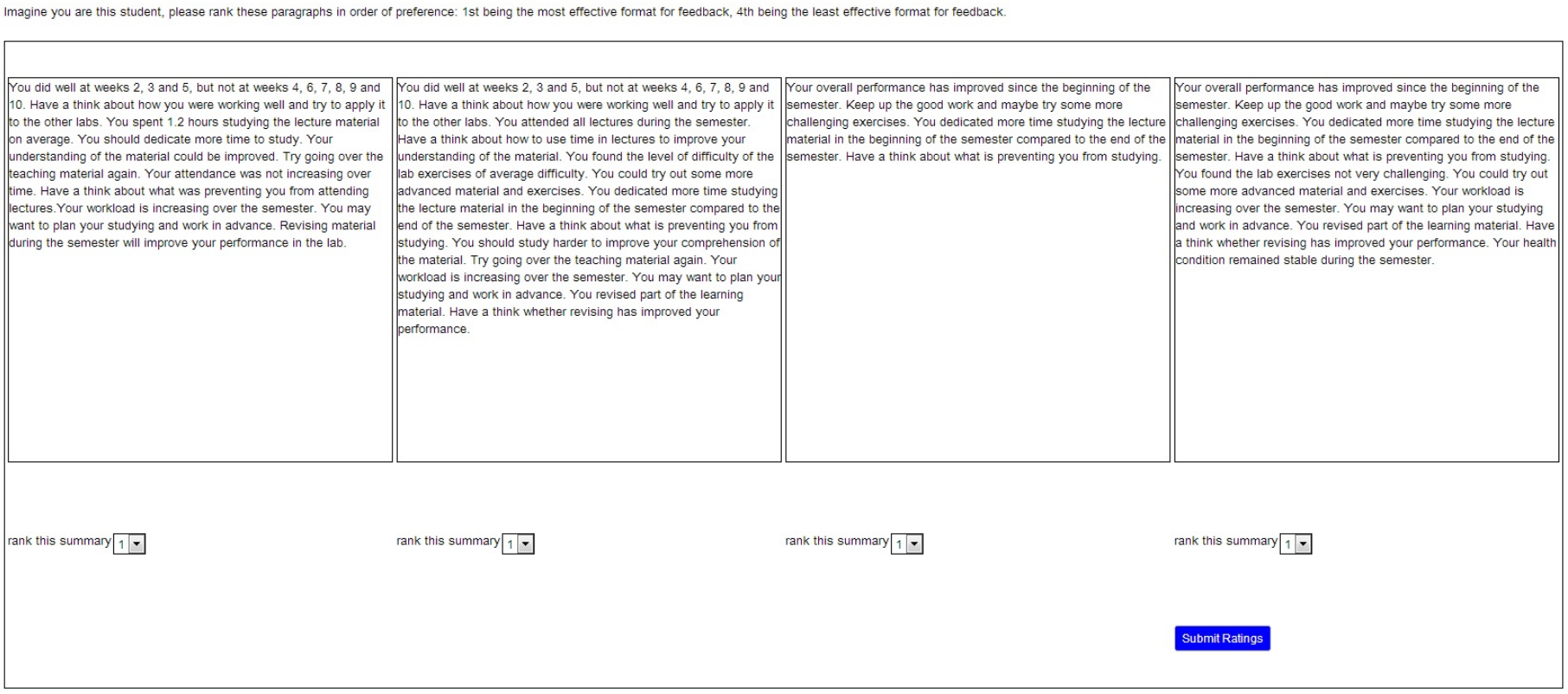


Figure 4.4: The interface for the evaluation: the students viewed the four feedback summaries and ranked them in order of preference. From left to right, the summaries as generated by: an Expert (Baseline 3), the Rule based system (Baseline 1), the Brute Force algorithm (Baseline 2), the RL system.

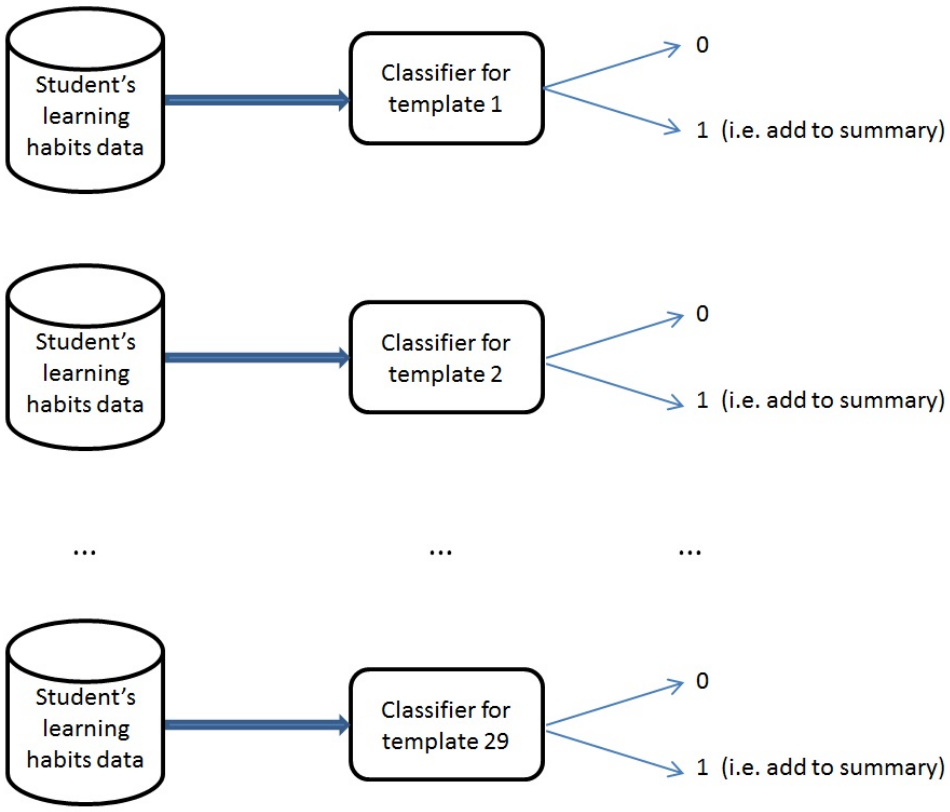


Figure 4.5: Feedback generation as a binary classification problem. 29 classifiers need to be trained, each one is responsible for each template. No history is taken into account.

Figure 4.7: The evaluation setup. Students were presented with the data in a graphical way and then they were asked to evaluate each summary on a 10-point Rating scale. Summaries displayed from left to right: ML system, RL, rule-based and random.

Chapter 5

Multi-adaptive Natural Language Generation

An `adaptive` Natural Language Generation system is able to generate text from non-linguistic data, ideally `adapting` the content to a specific user group. In some cases, there are multiple stakeholders with their own individual goals, needs or preferences, for example managers with employees or doctors with patients and relatives. In this chapter, we will investigate whether it is feasible to simultaneously adapt content to two different *known* types of stakeholders, as for instance speakers and hearers (RQ2). For feedback generation, lecturers can be thought of as "speakers", as they are the stakeholders who would traditionally provide feedback, and students can be thought of as "hearers", as they are the receivers of feedback. Speakers and hearers have generally different preferences. Preference elicitation is a bottleneck to many research areas that address intelligent systems, such as decision support systems, natural language interfaces, robotics etc. In this chapter, we initially acquire the preferences of two different types of stakeholders, lecturers and students and then we explore the feasibility of combining their preferences, when generating summaries in the context of student feedback generation.

As discussed in Chapter 3, various factors can influence students' learning, such as difficulty of the material (Person et al., 1995), workload (Craig et al., 2004), attendance

in lectures (Ames, 1992) etc. These factors change over time and can be interdependent. The different stakeholders (i.e. lecturers and students) have different perceptions regarding what constitutes effective feedback. In our domain, for instance, lecturers tend to comment on the hours that a student studied, whereas the students least prefer this content. Producing the same summary for two groups is important as it allows for shared context and meaningful further discussion and reduces development time. Therefore, when generating feedback, we should take into account all preferences in order to be able to produce feedback summaries that are acceptable by all stakeholders.

The production of natural language from data is mainly based on imitating how experts provide summaries of data or using aligned corpora to train generation algorithms. However, most of these approaches model only the speaker (provider of text), not the hearer (receiver of text) or the collaborative nature of communication. In addition, the developed approaches to data-to-text generation, as the ones discussed in Chapter 2, are evaluated with experts or against expert-generated text, rather than the end users. One consequence of this is the focus on speaker-adapted algorithms, which are also evaluated on the basis of how well they match experts' choices. In contrast, in human communication, language is hearer-oriented (Stedt, 2011): the speaker's content choices adapt to those of the hearer's.

In this chapter, we take a step back of current practices, and consider the scenario where end-users take part in the evaluation of our systems. Current methods have focused mainly in two directions: (1) re-producing experts' output, as in (Barzilay and Lapata, 2005; Angeli et al., 2010; Konstas and Lapata, 2012) and (2) adapting to users' without taking into account the experts' view as in (Mahamood and Reiter, 2011; Williams and Reiter, 2008). In the domain of student feedback, both the views of lecturers and students are important. Lecturers are the experts, who can provide feedback based on models from educational theory, whereas the students need to find the feedback fair, comprehensive and accurate. This chapter not only contributes a model for accounting simultaneously for speakers and hearers, but also evaluates the

designed system with both speakers and hearers.

In Chapter 4, it was shown that the lecturers and the students who participated in the user studies rated the generated feedback summaries differently based on the selected content. In addition, we discussed that students rated higher the summaries generated by the rule-based system compared to the other systems. In this rule-based system a students' preferences were taken into account, along with the expert's suggestions. In this chapter, we explore combining students' and lecturers' preferences in a data-driven system. For a feedback generation system, where lecturers and students should have access to the same information, there is need for feedback summaries that fulfill the preferences of both user groups. Taking into account these observations, we introduce a new challenge: *Multi-adaptive Natural Language Generation (MaNLG)*, which refers to the task of automatically adapting the output to all stakeholders simultaneously.

NLG systems that address more than one known user group have been previously thoroughly studied, focusing on users with different background knowledge, experience and anticipation from the systems. NLG systems can employ different versions of a system for each different user group as for instance the BabyTalk (BT) project (Gatt et al., 2009; Hunter et al., 2011; Mahamood and Reiter, 2011). The BT project uses NLG systems in a Neonatal Intensive Care Unit (NICU) environment to automatically provide reports to different stakeholders. For example, BT-nurse is addressed to nurses working in NICU whereas BT-family is addressed to the parents and relatives of the baby. NLG systems have used User Models (UMs) in order to adapt their linguistic output to individual users (Janarthanam and Lemon, 2010; Thompson et al., 2004; Zukerman and Litman, 2001). For instance, Janarthanam and Lemon (2010) propose a system that adapts the generated Referring Expressions to a users' skills and inferred knowledge. Reiter et al. (1999) use a rule-based approach that employs questionnaires to derive information about the user in order to personalise the output to each individual. Finally, Han et al. (2014) suggest the use of latent User Models for NLG. In the latter

framework, instead of directly seeking the users' preferences or the users' knowledge through questionnaires, the UMs are inferred through "hidden" information derived from sources such as *Google Analytics*. Our proposed approach makes a contribution to this area by adjusting the output to the preferences of more than one type of stakeholder, lecturers and students. Instead of developing different versions of a system or employing UMs, it investigates the modeling of the middle ground between the preferences of different potential user groups.

This chapter is organised as follows. Section 5.1 presents a preliminary experiment on multi-adaptive content selection. Section 5.2 presents a multi-adaptive methodology that uses Principal Component Regression (PCR) to hand-craft a reward function which is used to train an RL agent. Section 5.3 presents the evaluation of the PCR-based method. Section 5.4 discusses the results and finally, Section 5.5 concludes the chapter.

5.1 Multi-adaptive NLG as a Multi-objective Optimisation task

In this section, we explore a method that aims to adapt to lecturers' and students' preferences simultaneously, by aggregating two objective functions as in Equation 5.3: one that describes the lecturers' preferences as seen in Chapter 4 and one that describes the students' preferences. Learning algorithms can be divided into two categories: single-objective learning and multi-objective learning. Multi-objective optimisation (MOO) can be applied to situations where optimal decisions are sought in the presence of trade-offs between conflicting objectives Deb (2001). In the previous chapter, content selection is treated as a single-objective problem, where the objective is to maximise lecturers' ratings of feedback summaries. Generally, a single-objective optimisation problem can be formulated as follows:

$$\max f(x) \tag{5.1}$$

where f is a function that models the objective and the goal of the learning agent is to maximize the objective function.

When multiple objectives are present the optimisation problem can be formulated as a Multi-Objective Optimisation problem as follows:

$$\max[f_1(x), f_2(x), ..., f_n(x)] \tag{5.2}$$

where f_1, f_2, ... f_n are functions that describe conflicting objectives that need to be maximised (or minimised) simultaneously. There are two ways of learning to solve a multi-objective problem:

1. By scalarising the objectives, i.e. where the output of the objectives is aggregated as for instance,

$$\max[(1/n) * f_1(x) + (1/n) * f_2(x) + ... + (1/n) * f_n(x)] \tag{5.3}$$

   where, the weights $1/n$ is just an example. The developer can specify different values for the weights which are appropriate for a particular problem. For example, in a problem with 2 objectives the developer might want to give 90% emphasis on the first objective and 10% on the second. We will discuss this approach in Sections 5.1.1 and 5.2.

2. Pareto-based approaches, where instead of a single output the agent returns a set of optimal solutions rather than a single solution. We will discuss a Pareto-based approach in the next chapter.

In the next section, we present a preliminary experiment for multi-adaptive NLG. We investigate whether we can find middle ground by taking into account the preferences of speakers and hearers simultaneously.

5.1.1 Exploratory Experiment

In order to explore our research question, we initially demonstrate that lecturers and students rank summaries differently and that it is possible to model their preferences separately. We ask lecturers and students to compare and rank three summarisation systems:

1. (1) a Lecturer-adapted system, which adapts to lecturers' preferences and is described in detail in Section 5.2.1,
2. (2) a Student-adapted, which adapts to students' preferences and is described in detail in Section 5.2.2, and
3. (3) a multi-objective optimisation (MOO) system.

The three systems use the RL setup presented in Section 4.1. The only difference lies in the reward function used for training each RL agent. The RL approach was chosen over the supervised one, as supervised learning needs aligned corpora to learn from, therefore it would not be appropriate for this task, as students cannot provide written feedback summaries, which could otherwise be used for training a multi-label classifier.

We examine the weights derived from the multiple linear regression to determine the preferences of the different user groups (See also Appendix C). Overall, it was found that lecturers and students find different content useful. For instance, lecturers' most preferred content is `hours_studied`, therefore the reward function gives high scores to summaries that mention the hours that a student studied in all cases (i.e. when the `hours_studied` increased, decreased, or remained stable). This, however, does not factor heavily into the student's reward function.

The MOO reward function attempts to balance the preferences of the two user groups by aggregating the Lecturer-adapted and Student-adapted reward functions. For this MOO function, the coefficient for mentioning `health_issues` is also negative, however the other coefficients are smoothed providing neither strong negative nor

positive coefficients. This means that this function meets neither group's criteria. Alternative weights could be explored, however, there is no mechanism of choosing the weights in an informed way, thus the weights are specified by the researcher.

The two user groups significantly preferred the output of the system which is trained for their preferences (Mann-Whitney U and Wilcoxon signed-rank test, $p < 0.05$). Interestingly, lecturers find both the outputs produced by the Lecturer-adapted system and the Student-adapted system significantly preferable (Mann-Whitney U and Wilcoxon signed-rank test, $p < 0.05$) compared to the output produced by the MOO system. In contrast, students significantly prefer the output generated by the Student-adapted system over the other two. Finally, both user groups rate the MOO system 3rd, but there is no statistically significant difference between the student ratings for the MOO system and the Lecturer-adapted system. This result shows that students are not happy with the MOO output and that there is space for improvement. In the next section, we will consider a different reward function that uses less features than the systems described here, and therefore it models less noise. For instance, the student function contains negative coefficients for all hours_studied content (ranging from -1 to -57) whereas the lecturer function rewards the inclusion of this content (from 122 to 155). Due to the fact that the coefficients in lecturer function are much higher, the MOO output would refer to hours_studied (coefficients range from 49 to 51.5. This might be one reason that explains why students rank the summaries from the Lecturer-adapted and MOO systems similarly. For more details, please see (Gkatzia et al., 2014b).

Machine learning in high-dimensional feature spaces can be inefficient due to the computational cost of processing many dimensions; the presence of noisy and redundant features; and the "curse of dimensionality" (Handl and Knowles, 2008; Bellman, 1961), i.e. when the dimensionality increases, the volume of the space increases which leads to data sparsity. Therefore, dimensionality reduction techniques are often employed to address this issue. In the next section, we investigate an alternative method of reward function derivation using Principal Component Regression (PCR).

5.2 Hand-crafted Reward Function through Dimensionality Reduction

As previously discussed, the feedback summaries can be transformed into vectors that consist of 90 features or variables. Modelling high-dimensional feature spaces (more than 10 features) introduces noise. PCR is a method that combines Principal Component Analysis (PCA) (Jolliffe, 1982) with linear regression. PCA is a technique for reducing the dataset dimensionality while keeping as much of the variance as possible. In PCR, PCA is initially performed to identify the principal components; in our case, the factors that contribute the most to the variance. Then, regression is applied to these principal components to obtain a vector of estimated coefficients. Finally, this vector is transformed back into the general linear regression equation. We evaluate this approach against two systems, one Lecturer-adapted and one Student-adapted.

In order to derive a reward function that finds a balance between the lecturers' and students' preferences, we use PCR to reduce the dimensionality of the data and thus reduce the introduced noise. Through PCR, we are able to identify components of factors that are deemed important to both parties to be used in the reward function. The knowledge acquired through this process is used to hand-craft a reward function that is used for training an RL agent as described in the previous chapter. PCR was performed using SPSS and the feature engineering was based on the derived coefficients as described in 5.2.3.

5.2.1 System 1: Lecturer-adapted

The first system is the unmodified RL system described in the previous chapter in Section 4.1. The reward function is the cumulative function presented in Section 4.1.1:

$$Reward_{LECT} = intercept + \sum_{i=1}^{n} b_i * x_i + b_{90} * length \tag{5.4}$$

where $X = \{x_1, x_2, ..., x_n\}$ is the vector of combinations of the data trends observed

in the time-series data and a particular reference type of the factor. The value of x_i is given by the function:

$$x_i = \begin{cases} 1, & \text{if the combination of a factor trend and a particular} \\ & \text{reference type (e.g. average, trend) is included in the feedback} \\ 0, & \text{if not.} \end{cases} \quad (5.5)$$

The coefficients represent the preference level of a factor to be selected and how to convey them in the summary. Important factors are associated with high positive coefficients and the unimportant ones with negative coefficients. In the training phase, the agent selects a factor and then decides whether to talk about it or not. If it decides to refer to a factor, the selection of the template is performed deterministically, i.e. it selects the template that results in higher reward as discussed in Section 4.1.3. Length represents the number of factors selected for generation. The coefficient a, b and c can be found in Appendix C.

5.2.2 System 2: Student-adapted

We utilised the student ratings from the experiment presented in Section 4.3 in order to acquire knowledge of the students' preferences. The reward function used for training is of a similar style as the Lecturer-adapted reward function. Again, we transformed the rated summaries into vectors in order to feed a linear regression model. An example of a vector is: $\{x_1, x_2, x_3, ..., reward\} = \{0, 1, 0, ..., 95\}$. The weights of this function were derived by applying linear regression in a similar way as Walker et al. (2000) and Rieser and Lemon (2011). The Student-adapted function is the following function:

$$Reward_{STUDENT} = intercept_{st} + \sum_{i=1}^{n} w_i * x_i + w_{90} * length \quad (5.6)$$

where $X = \{x_1, x_2, ..., x_n\}$ is the vector of combinations of the data trends observed in the time-series data and a particular reference type of the factor. The value of x_i is

given by the function:

$$x_i = \begin{cases} 1, & \text{if the combination of a factor trend and a particular} \\ & \text{reference type (e.g. average, trend) is included in the feedback} \\ 0, & \text{if not.} \end{cases} \quad (5.7)$$

The $intercept_{st}$ and the coefficients w are given in Appendix C.

5.2.3 System 3: Multi-adaptive-PCR

In order to identify the users' preferences, we apply Principal Components Regression (PCR (Jolliffe, 1982)) analysis to both datasets that contain lecturers' and students' ratings. This enables us to identify the most important variables from the principal components, which can then constitute the features of a reward function. This hand-crafted reward function can be used for training the RL agent for summarisation of time-series data. After performing this analysis on both datasets (students and lecturers), the most common and important principal components (i.e. the ones that contribute the most to the variance) are chosen to be included in the reward function. 18 features were found to be important for the reward function (see Table 5.1) as they were the ones that contributed most to the variance (over 70% of the variance).

Specifically, the reward function is the following cumulative function:

$$Reward = intercept' + \sum_{i=1}^{m} s'_i * x'_i \quad (5.8)$$

where, $m = 18$, $X' = \{x'_1, x'_2, ..., x'_m\}$ describes the chosen combinations of the factor trends observed in the time-series data and a particular template (i.e. the way of mentioning a factor) and is determined by the Equation 5.7. The coefficients s were handcrafted. For training, the same process as before is followed. The coefficients represent the level of preference for a factor and the way it is conveyed in the summary.

x_n	**coefficient**	**factor**	**trend**	**way it is mentioned**	**student**	**lecturer**
(1) x_0	6.5	difficulty	increase	average	28	-81
(2) x_2	7	difficulty	decrease	average	-8	-77
(3) x_8	29	hours studied	decrease	average	-21	146
(4) x_{10}	7.8	hours studied	other	average	-13	155
(5) x_{11}	73	hours studied	other	trend	-9	205
(6) x_{13}	23	understandability	increase	trend	108	-3.3
(7) x_{29}	52	health issues	decrease	weeks	-17	5.49
(8) x_{31}	7.4	health issues	other	weeks	-25	40
(9) x_{35}	7	personal issues	decrease	weeks	-67	34
(10) x_{37}	2.2	personal issues	other	weeks	-99	41
(11) x_{38}	42	personal issues	other	trend	37	-10
(12) x_{43}	8.9	lectures attended	decrease	weeks	-10	1.6
(13) x_{45}	19	lectures attended	other	average	-50	28
(14) x_{46}	44	lectures attended	other	weeks	-25	20
(15) x_{49}	-21	revision	increase	other	-99	-88
(16) x_{52}	-77	revision	other	average	-89	-49
(17) x_{57}	39	marks	decrease	average	-68	92
(18) x_{60}	79	marks	other	average	-150	85

Table 5.1: The 18 features selected through PCR analysis.

For instance, if a coefficient has a high positive value, it is high likely that the content represented by this coefficient will be chosen for generation. Equally, if a coefficient has a low value, the corresponding content is less likely to be present in the summary. As such, the hand-crafted function can influence the content selection completely. The benefit of this approach is that the values of the coefficients can be easily modified, if the generation goal changes, for example if we want to adapt content to a specific user rather than a group of users. In this case, the coefficients that correspond to the users' most preferred can be set to high positive values.

5.3 Evaluation and Results

The multi-adaptive system is evaluated with students and lecturers against the Lecturer-adapted system and the Student-adapted system (as discussed in page 99). The output of the three systems was ranked by 13 lecturers and 30 computer science students

Student-adapted	Multi-adaptive-PCR	Lecturer-adapted
You **did well** at weeks 2, 3, 6, 8, 9 and 10, but not at weeks 4, 5 and 7. Have a think about how you were working well and try to apply it to the other labs. Your **attendance** was varying over the semester. Have a think about how to use time in lectures to improve your understanding of the material. You found the **lab exercises not very challenging**. You could try out some more advanced material and exercises. You dedicated **more time studying the lecture material** in the beginning of the semester compared to the end of the semester. Have a think about what is preventing you from studying. **Revising material** during the semester will improve your performance in the lab.	Your **overall performance** was very good during the semester. Keep up the good work and maybe try some more challenging exercises. You found **the lab exercises not very challenging**. You could try out some more advanced material and exercises. You dedicated more **time studying the lecture material** in the beginning of the semester compared to the end of the semester. Have a think about what is preventing you from studying. You have had **other deadlines** during weeks 6 and 8. You may want to plan your studying and work ahead.	Your **overall performance** was very good during the semester. Keep up the good work and maybe try some more challenging exercises. You found the lab exercises not very challenging. You could try out some more advanced material and exercises. You **dedicated more time studying** the lecture material in the beginning of the semester compared to the end of the semester. Have a think about what is preventing you from studying. You have had other deadlines during weeks 6 and 8. You may want to plan your studying and work ahead. You did not face any **health problems** during the semester. You did not face any **personal issues** during the semester.

Table 5.2: Example outputs from the three different systems (bold signifies the chosen template content).

from a variety of years of study. Time-series data of three students were presented on graphs to each participant, along with three feedback summaries (each one generated by a different system), in random order, and they are asked to rank them in terms of preference.

Table 5.2 shows three summaries that have been generated by the different systems. As we can see from Table 5.3, students significantly prefer the output of the system that is trained for their preferences. In contrast, students significantly least prefer the system that is trained for lecturers' preferences. Finally, they rank as second the system that captures the preferences of both lecturers and students, which suggests that it models the middle ground well between the preferences of two user groups. Significance testing

is done using a Mann Whitney U test (p <0.05) and Wilcoxon signed test (p <0.05), performing a pair-wise comparison (Multi-adaptive vs. Lecturer-adapted, not significant different. Multi-adaptive vs. Student-adapted, * at p = 0.03. Lecturer-adapted vs. Student-adapted, * at p = 0.009).

Summarisation Systems	**Lecturers' Ranking (Mean)**	**Students' Ranking (Mean)**
Lecturer-adapted	**1st (1.825)**	3rd* (2.09)
Student-adapted	3rd* (2.275)	**1st* (1.83)**
Multi-adaptive-PCR	**1st (1.9)**	2nd (2.07)

Table 5.3: Mode (mean) of the ratings for each user group. Mann-Whitney U and Wilcoxon signed-rank test, $p < 0.05$, when comparing each system to the multi-adaptive-PCR system).

5.4 Discussion

By reducing the dimensionality of the feature space, the regression is able to more accurately model the feedback summaries, as less noise is included in the model. The weights derived from the linear regression analysis vary from the Lecturer-adapted function to the Student-adapted function. For instance, the lecturers' most preferred content is `hours_studied`. This, however, does not factor heavily into the student's reward function, apart from the case where `hours_studied` are decreasing or remain stable (see also Table 5.1).

Students like reading about `personal_issues` when the number of issues they faced was increasing over the semester. On the other hand, lecturers find it useful to give advice to all students who faced personal issues during the semester, hence `personal_issues` are included in the top 18 features (Table 5.1). Moreover, students seem to mostly prefer a feedback summary that mentions the understandability of the material when it increases, which is positive feedback.

As reflected in Table 5.1, the analysis of PCR showed that both groups found it useful to refer to the average of marks when they remain stable. In addition, both

groups found understandability when it increases useful, for a variety of reasons, for example lecturers might find it useful to encourage students whereas students might prefer to receive positive feedback. Both groups also agree on `hours_studied` as described earlier. On the other hand, both groups find mentioning the students' difficulty when it decreases as positive.

5.5 Conclusion

This chapter concerned with the task of adapting to "hearers" and "speakers" simultaneously, as for instance lecturers and students. We presented a method that uses PCR to extract the most important preferences of the two groups and then hand-crafts a reward function based on this analysis.

In this chapter, we initially developed two functions that can model students' and lecturers' preferences respectively. These functions were used for the development of two systems that are optimised for each groups' preferences. The PCR-based approach was compared to these two systems. It is shown that the multi-adaptive-PCR method is consistent in stakeholders' preferences elicitation, through identification of the most relevant features that contribute to participants' high ratings. Because lecturers provide such different feedback summaries, it is expected that not all lecturers would rate a feedback summary similarly and potentially this is the reason why they rated the multi-adaptive-PCR system higher than the Lecturer-adapted. As the PCR-based system is thorough and carefully crafted, it produces high quality output and therefore the students rated it higher than the lecturer-adapted system.

This chapter also explained why a multi-objective approach which scalarises the two preference functions, is not able to tackle the challenge of multi-adaptation. Students and lecturers have conflicting preferences and the scalarisation leads to cancelling out the coefficients of both functions. In the next chapter, we will therefore investigate a Pareto-based approach. Not all domains have predefined user groups or users that have known group membership. We will explore handling unknown users, i.e. users

whose group membership in terms of preferences is *unknown*. Consider, for instance, a decision support system that generates textual summaries of physiological sensors in the context of first aid provision. The system should account for users with different background and preferences. As this is a time critical scenario, user modelling cannot be performed at the time of a casualty. We will show that, because an unknown user will belong to one of the already defined user groups, a multi-objective approach that is able to find optimal or near optimal summaries for all groups simultaneously will be generally preferable by unknown users. The health data described in Chapter 3 will be used.

Chapter 6

Accounting for Unknown Users using Multi-adaptive NLG

This chapter presents an approach to content selection from medical sensor data, which is able to handle *unknown* users. It investigates whether we can effectively address *unknown* users, i.e. users with unknown group membership (RQ3). As unknown users have unknown group membership, we argue that a multi-adaptive approach is able to handle them. In the previous chapter, MaNLG was used to simultaneously find the middle ground of the content preferences of lecturers and of students (i.e. known stakeholders), when generating feedback summaries. This chapter shows how this approach can be applied to account for unknown users / stakeholders in the context of a decision support system for first aid provision.

The contributions of this chapter to the field are:

1. It minimises regret for unknown users.
2. It employs a clustering approach for grouping users depending on content preferences rather than demographic qualities.
3. It presents a novel approach to Multi-adaptive NLG.

Many elements of a decision support system can be fixed, i.e. events, information sources etc. (Boutilier, 2013). What differs are the users' preferences. Unfortunately,

users' preferences cannot be inferred without prior interaction with or knowledge of the user. Our proposed methodology accounts for unknown users by minimising *regret*. Regret in decision theory, also known as opportunity loss, is defined as the difference between the actual payoff and the payoff that would have been gained if a different action had been chosen (Loomes and Sugden, 1982). In our domain, the first aider is unknown, as it is normally rare for someone to be requested to provide first aid on a regular basis. For example, in a hypothetical first aid scenario, the system should address users with different levels of expertise, from medical doctors to bystanders. In this scenario, time is critical and user profiling cannot be performed at the time of a casualty. For example, it would be inappropriate to ask users about their background at this point. As such, our system optimises the output with respect to a pool of potential users. Therefore, the derivation of the preferences of a particular user is difficult and thus we need to develop a system that is able to produce summaries of sensor data that are acceptable by all potential users. Other areas that could potentially benefit from such an approach are online content applications, travel apps, etc.

In Chapter 5, we introduced a PCR-based approach, which finds the most important features of a high dimensional feature space and then hand-crafts an optimisation function based on this analysis. The domain used here has a small feature space and therefore PCR cannot be used. To overcome this barrier, this chapter develops a Pareto-based approach to Multi-adaptive NLG that is able to handle small feature spaces using genetic algorithms. In this approach, instead of having a single output as a solution, the algorithm returns a set of optimal solutions (Pareto set) for each group. All solutions are pulled together and the one that is preferable to both groups is chosen. We will describe this approach in detail in Section 6.2.

In addition, instead of relying on predefined group definitions such as users' occupation or gender or role, as for instance lecturers and students (Gkatzia et al., 2014b) or doctors and nurses (Gatt et al., 2009), a cluster-based approach is used for defining the latent clusters regarding participants' template choices, following (Dethlefs et al.,

2014). We also show that this methodology which was initially developed for surface realisation in the restaurant recommendation domain (Dethlefs et al., 2014), is applicable to the content selection task in a different domain. For instance, this approach has impact not only on NLG, but also on interactive systems such as decision support systems. Boutilier (2013) refers to a related task as `preference aggregation` in the context of a group decision support system. In this framework, the users (or decision makers) rank the alternative decisions in terms of preference. Then, a voting rule is used to decide on the output. This framework assumes that all users have different preferences and thus the voting rule is essential. In contrast, we cluster users in terms of content choices and therefore we are able to model the middle ground between their preferences. We then treat the task of content selection as a multi-objective optimisation (MOO) task, where the preferences of each user group (cluster) are modelled as objective functions. As we discussed in Chapter 5, MOO refers to the task of optimising two or more objective functions simultaneously (Deb, 2001).

Section 6.1 briefly discusses the dataset collected in Chapter 3. Section 6.2 describes the overall methodology. Section 6.3 presents the evaluation setup, Section 6.4 reports and discusses the results obtained. Finally, Section 6.5 summarizes the chapter.

6.1 Data

The dataset consists of 314 instances collected by 70 participants with various levels of expertise, ranging from medical doctors to people with no prior experience or training in first aid provision. As a result, each instance in the dataset consists of a scenario (e.g. Figure 6.1), measurements of three sensors (e.g. Figure 6.2) and three selected templates from the list of available templates (e.g. Table 3.3.3). The templates were chosen by the participants in terms of preference. The template choices correspond to the preferred content of each user.

Scenario:

A female aged 30 years has been rescued from a burning building by Fire Service personnel. She is conscious and breathing. She has no obvious burns but is suffering from smoke inhalation and is currently being treated with 100 % oxygen by fire crews. The following graphs show the measurements of her breathing rate, blood oxygen saturation and heart rate. The summary below describes the sensor data depicted on the graphs. Please rate the summary in terms of your preference.

Figure 6.1: First Aid Scenario

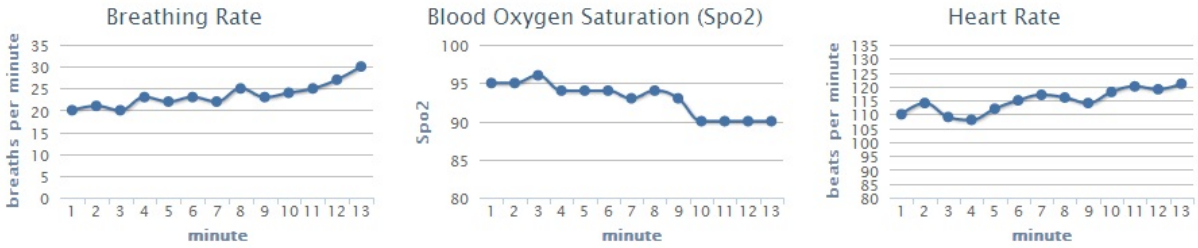


Figure 6.2: Physiological time-series data on charts

6.2 Methodology

The methodology consists of four steps as depicted in Figure 6.3:

1. We clustered users in terms of template choices using the Expectation-Maximisation (EM) algorithm, such that users with similar preferences belong to the same cluster (Section 6.2.1). The number of underlying clusters is unknown therefore EM is used, because it is able to automatically determine the number of underlying clusters. This facilitates the accurate modelling of user preferences in each cluster.

2. Having partitioned the users into two clusters, we derive two objective functions based on the participants' preferences in each cluster (Section 6.2.2), using *logistic regression*.

3. The two objective functions were used in a multi-objective optimisation framework in order to derive a solution (a summary) that is preferable by both. We used genetic algorithms (Section 6.2.3) to solve this MOO task which is the standard approach in this area (Deb, 2001).

4. This framework outputs a set of optimal solutions, known as a *Pareto set*, rather

than a single solution. All solutions are ranked regarding their scores by both objective functions which facilitates the selection of one solution, as we describe later (Section 6.2.4).

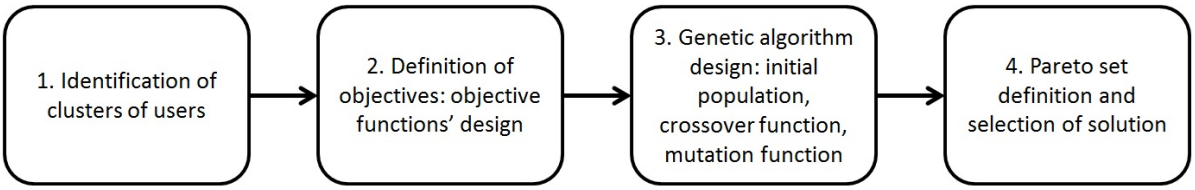


Figure 6.3: Methodology for addressing unknown users.

6.2.1 Cluster Analysis

The results in Chapter 3 showed that individual user characteristics, such as medical training level, gender, or experience with medical sensor data, do not have a significant effect on the template choice. The only significant factors are scenario and physiological parameters: Breathing Rate (BR), Blood Oxygen Saturation (SpO_2) and Heart Rate(HR). We conclude that categorising users depending on their pre-hospital training level, or their gender, or prior experience with sensor data, does not necessarily yield distinctive user groups. For instance, users that have received training at work can have similar preferences to medical doctors. We therefore consider automatic clustering to define user groups in terms of phrase choice, regardless of their training background, gender or experience with sensors, following a similar approach to the one presented by Dethlefs et al. (2014). However, their approach addresses known users, in the sense that the user preferences are defined via previous ratings on generated text. In comparison here, we deal with unknown users and therefore placing a user into a group is not possible. Cluster analysis allows one to group a set of objects in such a way that objects in the same group (cluster) are more similar (here in terms of their phrase choice) to each other than to those in other groups/ clusters. For instance, people who prefer referring to the average value of time-series are more similar and thus they belong to the same cluster, whereas people that prefer to refer to the trend in a verbose way belong to a

different cluster, etc. In this way, users are grouped according to their preferences and regardless of their profession, gender, or level of training.

We applied the Expectation-Maximization (EM) clustering algorithm using the WEKA toolkit (Witten and Frank, 2005). EM is useful when the number of the clusters is *unknown* (or generally not obvious), as in our dataset. Consequently, we need a clustering algorithm that is able to *determine the number of clusters automatically*. EM initially assigns a probability distribution to each instance which indicates the probability of it belonging to each of the clusters. It uses cross validation to determine the number of clusters following these steps: (1) sets the number of clusters to 1; (2) splits the training set into 10 folds randomly; (3) EM algorithm is applied 10 times as normally in cross validation; (4) it averages the log likelihood over all 10 results; and (5) if log likelihood has increased, the number of clusters is also increased by 1 and the procedure is repeated until convergence is achieved. EM works as follows:

Algorithm 5: EM clustering algorithm.

Input : Set of features X, set of n instances on X
Output: numOfClusters, clustered instances
Calculate the cluster probabilities for each instance;
numOfClusters=1;
Split dataset into 10 folds;
Apply EM to each fold;
Average log_likelihood over 10 folds;
if *i=0... m* **then**
    Select random k-labelset from L;
    Train an LP on D;
    Add LP to ensemble;

Our clustering task was formed as follows: given the choices that a participant makes over all scenarios, assign the user into a group. Accordingly, the features used for clustering are all the template choices a user makes for all four scenarios. EM clusters the data in two consistent user groups, where the first cluster consists of 27 participants and the second consists of 43 participants. Figure 6.5 shows the distribution of members of a group (as per level of training) for each cluster. We use *Chi-squared test* to evaluate the consistency of the clusters in terms of the scenarios and the template choice. We

also notice that the scenarios and the time-series data are multicollinear[9], i.e. the scenarios and the time-series data are highly correlated. Therefore, the analysis using the scenarios will produce exactly the same results as if we use the trends of time-series data instead. In the following sections the two clusters are discussed in detail.

Scenario	**Template**	**Breathing rate**	**SpO2**	**Heart rate**
1. Smoke inhalation	(1) Average	0%	0%	0%
BR: incr	(2) Trend verbose	33.2%	25.9%	3.75%
SpO2: decr	(3) Trend succinct	**51.8%**	**66.6%**	**92.5%**
HR: incr	(4) Range verbose	0%	3.75%	3.75%
	(5) Range succinct	15%	0%	0%
	(6) Inference	0%	3.75%	0%
2. Drowning	(1) Average	24%	20%	0%
BR: stable	(2) Trend verbose	0%	4%	16%
SpO2: stable	(3) Trend succinct	8%	12%	**72%**
HR: incr	(4) Range verbose	16%	24%	0%
	(5) Range succinct	**40%**	**36%**	4%
	(6) Inference	12%	4%	8%
3. Falling down stairs	(1) Average	16.6%	12.5%	0%
BR: stable	(2) Trend verbose	4.2%	0%	8.3%
SpO2: stable	(3) Trend succinct	**37.5%**	12.5%	**83.3%**
HR: decr	(4) Range verbose	8.35%	25%	0%
	(5) Range succinct	25%	**45.8%**	4.2%
	(6) Inference	8.35%	4.2%	4.2%
4. Bicycle accident	(1) Average	4.2%	16.6%	0%
BR: incr	(2) Trend verbose	12.5%	0%	4.2%
SpO2: stable	(3) Trend succinct	**66.6%**	25%	**83.3%**
HR: incr	(4) Range verbose	8.35%	20.8%	0%
	(5) Range succinct	8.35%	**37.6%**	4.2%
	(6) Inference	0%	0%	8.3%

Table 6.1: The phrase frequencies (%) of each scenario for Cluster 1.

Analysis of Cluster 1

The first cluster consists of 10 male and 17 female participants (Figure 6.4). Regarding expertise, one participant belongs to Group 1, 17 to Group 2, one to Group 3, none to Group 4, three to Group 5 and five to Group 6 (Figure 6.5). For a description of each

[9]Multicollinearity in statistics exists when two or more explanatory variables are highly correlated.

group, please see Table 3.6. Nine participants have previous experience with sensor data and 18 do not (Figure 6.7). Table 6.1 shows the phrase choice frequencies for each scenario for Cluster 1. It is clear that participants in Cluster 1 generally prefer the succinct ways of referring to time-series data. Table 6.1 shows in detail the preferences of the users in Cluster 1.

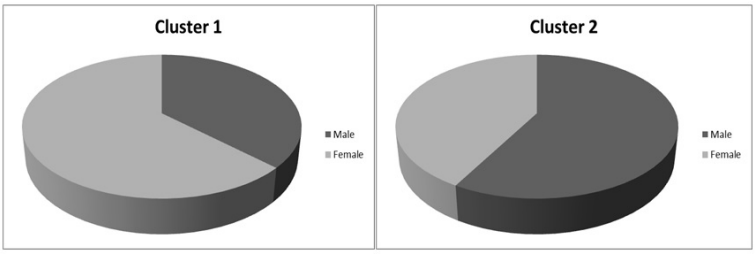


Figure 6.4: Males/Females in Cluster 1 and Cluster 2.

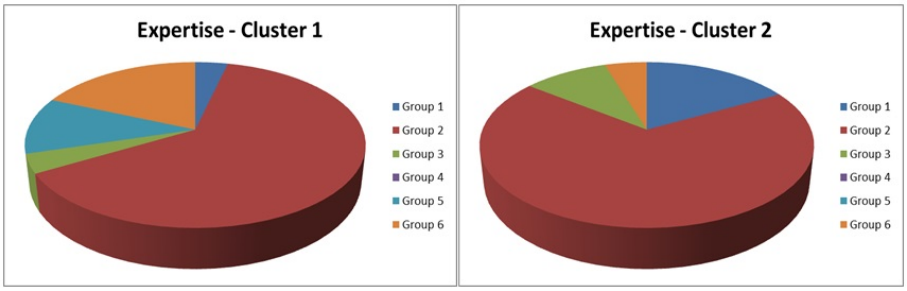


Figure 6.5: Different levels of expertise in the two clusters.

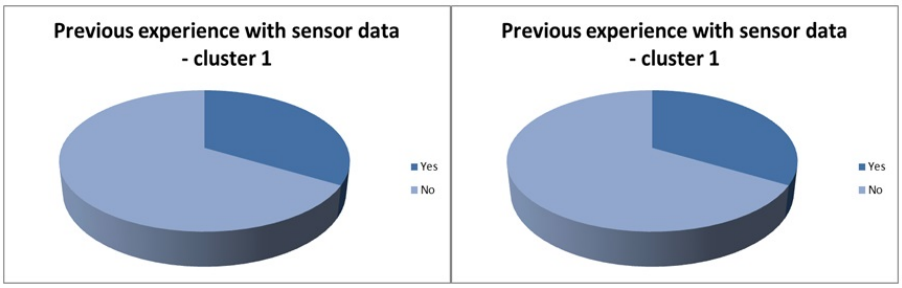


Figure 6.6: Previous experience with sensor data in the two clusters.

Analysis of Cluster 2

The second cluster consists of 25 male and 18 female participants (Figure 6.4). Regarding expertise, eight participants belong to Group 1, 27 to Group 2, four to Group 3,

none to Group 4, none to Group 5 and three to Group 6. Six participants have previous experience with sensor data and 37 do not (Figure 6.7). Table 6.2 shows the phrase choice frequencies for each scenario for cluster 2. In contrast with cluster 1, we observe that the users in this cluster prefer the verbose ways of referring to time-series data.

In particular, we see that the users in Cluster 2 prefer mentioning the trend for all variables in a verbose way in the `smoke inhalation scenario`. In the `drowning scenario`, the users prefer the sentences that describe the range of values for the Breathing Rate and Blood Oxygen Saturation, where these two variables remain stable throughout the monitoring. Users in Cluster 1 also prefer to mention the range of values, but in a succinct way. When the Heart Rate variable increases, the users prefer to refer to the trend. Similar preferences are observed for the other two scenarios.

In conclusion, it is obvious that the two clusters are very similar in that for the same scenarios, all users seem to agree on the way that they would refer to time-series data (e.g. referring to trend over the range).

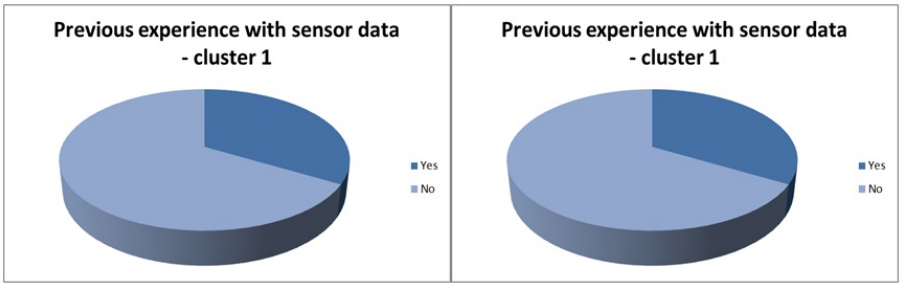


Figure 6.7: Previous experience with sensor data in the two clusters.

6.2.2 Preference Elicitation

Having defined the clusters as discussed previously, the next step is to acquire the preferences of each cluster. We used *iterated logistic regression* to estimate the probability of each template (see also Table 3.3.3) to be chosen (content selection decision) given the previous decisions. In previous work, linear regression is used to derive a model that can predict users' rating and thus maximise it (Walker et al., 2000; Rieser and Lemon, 2011). Linear regression assumes that there is a linear relationship between the

Scenario	Template	Breathing rate	SpO2	Heart rate
1. Smoke inhalation	(1) Average	2.3%	0%	2.4%
BR: incr	(2) Trend verbose	**93.1%**	**93.1%**	**79.0%**
SpO2: decr	(3) Trend succinct	0%	0%	0%
HR: incr	(4) Range verbose	2.3%	4.6%	16.2%
	(5) Range succinct	0%	0%	0%
	(6) Inference	2.3%	2.3%	2.4%
2. Drowning	(1) Average	34.1%	39%	0%
BR: stable	(2) Trend verbose	4.8%	0%	**97.5%**
SpO2: stable	(3) Trend succinct	0%	0%	0%
HR: incr	(4) Range verbose	**41.5%**	**58.5%**	0%
	(5) Range succinct	2.5%	2.5%	0%
	(6) Inference	17.1%	0%	2.5%
3. Falling down stairs	(1) Average	23%	38.4%	0%
BR: stable	(2) Trend verbose	**35.9%**	0%	**87.2%**
SpO2: stable	(3) Trend succinct	0%	0%	2.6%
HR: decr	(4) Range verbose	20.6%	**59%**	7.6%
	(5) Range succinct	2.6%	2.6%	0%
	(6) Inference	17.9%	0%	2.6%
4. Bicycle accident	(1) Average	10.6%	39.4%	0%
BR: incr	(2) Trend verbose	**73.7%**	2.7%	**97.3%**
SpO2: stable	(3) Trend succinct	0%	0%	0%
HR: incr	(4) Range verbose	15.7%	**55.2%**	0%
	(5) Range succinct	0%	2.7%	0%
	(6) Inference	0%	0%	2.7%

Table 6.2: The phrase frequencies of each scenario for Cluster 2.

dependent and the independent variables, which is untrue for our domain. In contrast, logistic regression estimates the probability of an event/decision occurring. In our task, each logistic regression model estimates the probabilities of a specific template to be chosen for generation given the time-series data.

6.2.3 Content Selection as a Multi-objective Optimisation Task

The standard method of solving multi-objective optimisation problems is through *genetic algorithms* (Deb, 2001). Genetic algorithms are inspired from the *Darwinian theory* of evolution, which states that the fittest organisms in nature are more likely

to be reproduced (Darwin, 1909). A genetic algorithm designed for MOO consists of (a) *a fitness function*, which is essentially the objective to be optimised (in the case of multi-objective optimisation there are two or more fitness or objective functions); (b) *a population* of chromosomes, which is a set of solutions, (c) *a ranking method*, which determines which chromosomes are to be selected for reproduction and (d) *genetic operators for reproduction*, which determine how the population evolves through mutation and/or crossover. Duboue and McKeown (2003) have also used genetic algorithms to derive content selection rules, but in a single optimisation framework.

6.2.3.1 Fitness (or Objective) Functions

We use multiple logistic regressions to calculate the probability of each template to be selected, given the previous decisions when possible. For instance, we know that the first decision to be made is the selection of a template that describes the BR. The next decision would affect the template that describes the SpO_2. Therefore, in our model, we include the decision made for the BR template as feature, when applying logistic regression for SpO_2. Similarly, the regression model for the HR decision, includes both the decisions made for the BR and SpO_2 template. Our decision to include the previous templates as features is motivated by the work presented in Chapter 5, which states that when summarising time-series data, current decisions about the content are influenced by the previous decisions.

The result of the logistic regressions is the probability of an event occurring. Intuitively, the objective function could not be other than the conditional probability of three templates occurring together. The goal is to maximise the conditional probability of t_{BR_i}, $t_{SpO_{2_i}}$ and t_{HR_i}, where t_{BR_i} is the template that describes the BR, $t_{SpO_{2_i}}$ is the template that describes the SpO_2 and finally the t_{HR_i} is the template that describes

the HR. The fitness function for the preferences of the first cluster can be written as:

$$Fitness(cluster_1) = \arg\max P(t_{BR_i} \cap t_{SpO_{2_i}} \cap t_{HR_i}) \tag{6.1}$$

In a similar way a fitness function was designed for the second cluster.

6.2.3.2 Population

Every possible summary can be encoded as a `chromosome`. A population consists of a set of chromosomes, i.e. a set of summaries. We encode chromosomes as a vector of 18 features, i.e. each feature corresponds to one template. Each chromosome consists of three genes corresponding to BR, SpO_2 and HR. Each gene consists of six binary features, each one describing a template type, as described in Chapter 3. An example of a chromosome for the scenario in Figure 6.1 can be:

$$\{0,1,0,0,0,0\},\{0,1,0,0,0,0\},\{0,1,0,0,0,0\}$$

which corresponds to the following summary:

> The breathing rate increased from 20 to 30 breaths per minute. The Blood oxygen saturation dropped from 95% to 90%. The heart rate increased from 110 to 121 beats per minute.

The initial population is randomly generated and its size is 20 (it was specified empirically), and it is kept similar for the next generations.

6.2.3.3 Ranking Method

We used the *maximum ranking* method to rank the chromosomes, similar to Schaffer (1985). The initial population was sorted in two lists in terms of each fitness func-

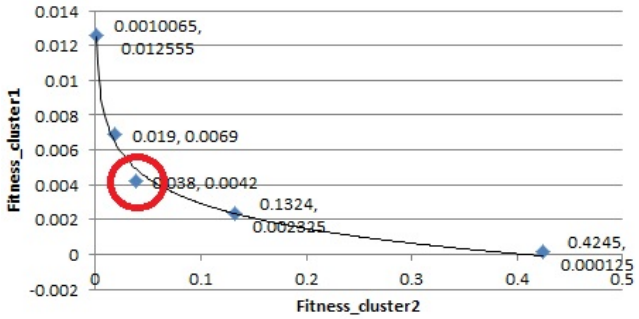


Figure 6.8: Chromosomes plotted on a graph. The red circle indicates the knee. The chromosomes that scored high only with one function are omitted in order to make the graph clearer.

tion. Then the eight fittest chromosomes were chosen from each list. Two additional chromosomes were also chosen randomly from each list to increase diversity.

6.2.3.4 Genetic Operators for Reproduction

For reproduction, ten chromosomes (parents) were randomly chosen; the nine first chromosomes and one chromosome was randomly chosen from the rest of the chromosomes. Ten new chromosomes are reproduced via *one-point crossover*. One-point crossover is the process where a single crossover point is chosen and all the data before this point adopt the genes from the first parent and beyond this point from the second parent and vice versa (so as two chromosomes are reproduced by a pair of parents). Then, five chromosomes are randomly chosen from the population and are *mutated* (only one gene is randomly changed). The same process continues iteratively until the stopping criterion is met, i.e. when there is no improvement in terms of fitness of the top (most optimal) chromosome. The method is chosen empirically.

6.2.4 Choice of Optimal Solution

The choice of the unique solution from the Pareto set is based on the *knee* approach (Branke et al., 2004; Handl and Knowles, 2008), also known as the elbow approach. The idea is that the solution located in the knee, when plotting the solutions on a graph, scores well for all objectives, as for instance in Figure 6.8. This approach ranks

Cluster1-based	**Cluster2-based**	**Multi-objective Optimisation (MOO)**
Resps ↑ from 20 to 30. SATS ↓ from 95% to 90%. Heart rate ↑ from 110 to 121.	The breathing rate **increased** from 20 to 30 breaths per minute. The Blood oxygen saturation was **between** 90% and 95%. The heart rate **increased** from 110 to 121 beats per minute.	The breathing rate **increased** from 20 to 30 breaths per minute. The Blood oxygen saturation **dropped** from 95% to 90%. The heart rate **increased** from 110 to 121 beats per minute.

Table 6.3: Example outputs from the three different systems (bold signifies the chosen template content).

the candidate summaries in terms of fitness for Cluster 1 and Cluster 2. It then chooses the summary that, for both fitness functions, the summary does not score worse than most of the other summaries.

6.3 Evaluation

In order to evaluate our methodology, the output of the MOO system is compared in a human evaluation against two meaningful baselines:

1. **Cluster1-based** optimises the content for the first cluster.

2. **Cluster2-based** is optimised for the second cluster. This baseline roughly corresponds to the *majority baseline*.

Example outputs of the three systems are shown in Table 6.3 for the scenario in Figure 6.1. It is obvious that the Cluster1-based function chooses templates that describe the content in a succinct way, which aligns with the analysis of Cluster 1. Cluster2-based and the MOO generate verbose output. We observe that the MOO system refers to the trend for all three measurements which is a common feature for both clusters.

For the evaluation, 21 new participants were recruited. Similarly to our setup for the initial data collection (Chapter 3), each participant is presented with an emergency scenario, i.e. graphs that depicted the physiological data gathered from the person in

need. This time participants are also presented with a summary of the time-series data as generated by one of our systems. They are asked to rate the summary on a 5-point Likert scale (*Dislike, Slightly dislike, Neither like nor dislike, Like overall/it's ok, Like very much*). The participants repeat this process three times for each of the three different scenarios. For each scenario they are presented with a summary generated by a different system, as Table 6.3 shows.

6.4 Results

Table 6.4 shows the mean, mode and standard deviation of the human ratings. Results from a pair-wise Mann-Whitney U test are shown in Table 6.5 along with the effect size (Cohen's d).

System	**Mean**	**Mode**	**Standard deviation**
MOO	3.75	4	0.89
Cluster1-based	2.9	2	1.17
Cluster2-based	3.22	4	1.34

Table 6.4: Mean, mode and standard deviation of user ratings.

Systems	**p-value**	Effect size
MOO vs. Cluster1-based	0.012*	0.796
MOO vs. Cluster2-based	0.24	0.461
Cluster1-based vs. Cluster2-based	0.356	0.246

Table 6.5: Significance (at $p < 0.05$) is indicated as * as determined by a Mann Whitney U test and effect size (Cohen's d) for pair-wise comparison.

The participants rate the output from the MOO system higher than the other two systems. In particular, participants statistically significantly prefer the MOO system to Cluster1-based system ($p < 0.05$). They also rate the MOO system higher than the Cluster2-based system, although not statistically significantly. As the statistical significance examines whether the results are likely to be due to the chance Sullivan and

Feinn (2012), we also report the effect size in order to understand the magnitude of the differences found. The effect size of the difference between the MOO and Cluster1-based is large (≈ 0.8). There is also medium effect between the MOO and Cluster2-based system ($= 0.461$). The effect size for Cluster2-based and Cluster1-based systems is small ($= 0.246$). Between the two latter systems, there is a tendency of ranking the Cluster2-based system higher than the Cluster1-based system. We assume that this is due to the fact that the majority of users tend to belong to Cluster2.

The standard deviation of the ratings indicates that those for the MOO system are more consistent compared to the other two systems. This shows that most users rate the MOO system consistently higher. The Cluster2-based system has the highest standard deviation, which means that the ratings of this system are more variable, i.e. Cluster-1 users would rate this cluster lower.

As the MOO system and the Cluster2-based system are similarly rated by the participants, we conclude that our MOO-based approach is able to simultaneously satisfy user preferences for *unknown* users, i.e. users who potentially belong to any of the clusters.

6.5 Conclusions

This chapter presents a novel content selection approach for a Natural Language Generation system, using multi-objective optimisation to address unknown prospective users. The chapter makes three contributions: (1) it minimises regret for unknown users, (2) it develops a novel multi-objective optimisation approach to multi-adaptive NLG, and (3) it transfers the cluster-based approach presented by Dethlefs et al. (2014) to a new task (content selection) and new domain (health informatics).

Generally, there are two approaches to multi-objective optimisation. The first approach combines the two objective functions into a single function, for example through a weighted sum. A common issue of this approach is choosing the appropriate weights. The 2nd approach generates a set of potential summaries which are ranked with re-

spect to their scores obtained by the objective functions and chooses the summary that does not score worse than most of the other solutions with respect to both functions. We showed in this chapter that such an approach generates summaries preferable by prospective users, irrespective of group membership. This allows us to minimise regret with respect to *unknown* users, i.e. users for whom we have no explicit information about their preferences. This data-driven approach is domain and task general and could be used in other content-based applications, such as interactive systems and decision support systems.

Chapter 7

Conclusions and Future Directions

This thesis has developed and evaluated approaches to content selection for adaptive and non-adaptive data-to-text systems. This chapter summarises the main contributions and findings in Section 7.1 and it indicates possible avenues for future work in Section 7.2.

7.1 Contributions and Findings

This section is broken down into three subsections with alignment to the research questions presented in the introductory chapter.

RQ1: With respect to user preferences, can the task of content selection be formulated and solved effectively using different data-driven techniques and hand crafted methods?

This thesis initially investigated data-driven approaches (Chapter 4) to content selection with respect to users' preferences. It developed, compared and evaluated two novel content selection methods with several baselines and made the following contributions:

- It contributed a novel and efficient method for tackling the challenge of content selection using a Reinforcement Learning (RL) approach (Section 4.1). The content selection task was formulated as a Markov Decision Process and Reinforcement

Learning was used for solving it, following previous work by Rieser and Lemon (2011). To our knowledge, this is the first effort of applying RL to a data-to-text application. The output of the RL system was preliminary evaluated in simulation and with students against the output of (1) a rule-based system; (2) a Brute-Force system; (3) lecturer-constructed summaries; and (4) a random system. It was found that students rated the output produced by the RL system comparable to the Lecturer-constructed summaries.

- The second approach treats the content selection task as a classification task (Section 4.2). This thesis presented an innovative supervised learning approach to content selection using Multi-label classification (MLC). MLC is able to capture dependencies between the data as it makes the decisions for content selection simultaneously. In addition, due to its nature, it can handle training data that display variability; for instance, when different experts provide different summaries for the same data. MLC is able to handle such datasets due to its nature. A comparison to 3 binary classification methods (Section 4.2.2) was presented. It was found that MLC is able to achieve higher Accuracy, Precision, Recall and F-score in 10-fold validation than the other classification methods.

- Finally, in Section 4.3, a comparison of MLC with RL was presented. The comparison showed that each method can serve different goals. MLC performed significantly better in automatic evaluation whereas RL performed better in human evaluation.

From the results, we draw the following three conclusions:

1. RL approaches can lead to optimised content selection and thus increased user ratings. RL is also able to generalise over unseen scenarios.

2. Multi-label classification is able to capture dependencies between the data and thus produce output similar to the gold standard, i.e./ output observed in the training data.

3. From the comparison of the two approaches, we learn that user preferences should also be taken into account for generation.

4. Because RL can optimise the output according to a reward function and can be used for automatically adapting the output to specific users.

5. MLC can generally be used when the aim of the generation is to replicate phenomena seen in the dataset, because it achieves high accuracy, precision and recall.

6. RL is more flexible and provides more varied output.

RQ2: Can we simultaneously adapt content to different *known* stakeholders?

NLG systems are often developed with the assistance of domain experts, however the end users are normally non-experts. Consider for instance a student feedback generation system, where the system imitates the teachers, but the end users are the students. The system will produce feedback based on the lecturers' rather than the students' preferences. Therefore, this thesis investigated whether it is possible to adapt the content to "speakers" and "hearers" simultaneously, i.e. to lecturers and students (Chapter 5).

The main contributions resulted from the second research question are the following:

- There are two types of *known* stakeholders, lecturers and students, in the domain of student feedback generation. Initially, we developed and presented two models that describe each groups' preferences respectively by exploiting lecturers' and students' ratings respectively. Using these functions as reward function for RL systems, we developed two systems, one that adapts to lecturers and one that adapts to students.

- We further utilised the user ratings to develop a novel approach that analysed the preferences of the two groups using Principal Component Regression and used the derived knowledge to hand-craft a reward function that is then optimised by an RL system. The results show that the end users prefer the output generated

by this system, rather than the output that is generated by a system that mimics the experts.

The following conclusions are drawn:

1. It is possible to model the middle ground of the preferences of different *known* stakeholders.

2. Optimisation of end users' preferences can lead to preferable output.

3. Because experts provide quite variable feedback summaries, it is hard to reach agreement between all experts. Therefore, experts, rated the PCR-based system similarly to the Lecturer-adapted system.

4. Majority-based baselines cannot optimise for user preferences because they cancel out conflicting preferences.

RQ3: Can we effectively address *unknown* users or stakeholders, i.e. users with unknown preferences or group membership?

In most real world application first-time users are generally *unknown*, which is a common problem for NLG and interactive systems: the system cannot adapt to user preferences without prior knowledge. This thesis finally developed a novel framework for addressing *unknown* first-time users, using Multi-objective Optimisation to minimise regret for multiple possible user types (Chapter 6). In this framework, the content preferences of potential users are modelled as objective functions, which are simultaneously optimised using Multi-objective Optimisation. The following contributions have been made:

- The developed framework helps minimising regret for unknown users.

- It employs a clustering approach that automatically determines the number of clusters depending on content preferences rather than demographic qualities.

- It presents a novel approach to Multi-adaptive NLG.

This work made the following observations:

1. In some domains, users should not be clustered in terms of demographic details or background knowledge. Instead, prospective users should be grouped in terms of preferences.

2. Users' preferences should be taken into account when grouping users, as their preferences can be independent of their background, job etc.

3. By finding common ground between the preferences of different groups of people, we can minimise regret.

4. It is possible to tackle the problem of first-time users by using Multi-objective optimisation. By optimising for all possible groups of users simultaneously, we consequently optimise for new users, as they will normally belong to one of the predefined clusters.

7.1.1 Discussion

The nature of data-to-text systems makes it difficult to directly compare end-to-end systems and algorithms in similar datasets and setups. In contrast to other NLG tasks, data-to-text generation is context-sensitive and therefore, previous work discussed in Chapter 2 has not been thoroughly compared with each other and the algorithm selection is usually based on intuition and theoretical foundation. Nevertheless, lessons learnt can be transferred across domains and contexts.

7.2 Future Work

The work presented in this thesis can be extended in various ways:

- **Student Performance as a Reward Function:** In this thesis, we used a preference-based reward function to train an RL agent and the developed systems

were evaluated in terms of user preferences (Chapter 4). In a different setup, the goal to be optimised would be the students' performance. In such setup, the system would receive feedback based on the marks the students achieved. Such system would optimise for task success.

- **Different Features as a Reward Function:** This thesis used the students' learning factors as features. A different approach would also consider their preferences on linguistic factors, such as lexical choices, motivational phrases, syntactic features etc.

- **Comparing the approach presented in Chapter 6 to the approach presented in Chapter 5:** The multi-objective approach presented in Chapter 6 could be transferred to the domain of student feedback generation and be compared to the PCR-based system.

- **Clustering students and lecturers in terms of preferences not in terms of role:** Chapter 5 discussed the differences between lecturers' and students' preferences and developed an approach for finding the common ground between those. However, this chapter did not look into differences in preferences in the same group. A more thorough study might disclose that some students prefer feedback from specific lecturers. Personalising feedback for each student could yield better results.

- **Transferring of the multi-objective approach to a new domain:** Multi-objective optimisation was used in this thesis to simultaneously optimise for the preferences of prospective first-time users that belong to different clusters. This approach could be transferred to the restaurant recommendation domain and extend Dethlefs at al. (2014) approach. Dethlefs et al. (2014) presented an approach that uses linguistic features to predict users' ratings on generated utterances, given only a couple of initial ratings. Our approach could extend this work in two ways:

  1. The clustering method presented in Chapter 6 automatically determines the

number of clusters. Dethlefs et al. (2014) experiment with different numbers of clusters instead. Adopting our approach would make their methodology straightforward.

2. Our approach could be extended to account for more than two clusters as is the restaurant recommendation domain.

3. Our approach could be used for generating the first couple of utterances needed for Dethlefs et al. approach. This way, the users satisfaction will be increased from the beginning of the interaction.

7.3 Conclusions

This chapter summarised the work presented in this thesis and discussed the contributions made. With respect to content selection, it also drew implications for adaptive systems. Finally, it presented directions for future work.

Appendix A

Feedback Generation: Templates

Each template is a quadruple consisting of an *id*, a *factor*, a *reference type* (trend, weeks, average, other) and *surface text*. An exhaustive list of the templates can be seen in the table below.

ID	**Factor**	**Type of template**	**Surface text**
1	difficulty	average	"You found the lab exercises *very/not so/not very* challenging. *Make sure that you have understood the taught material and don't hesitate to ask for clarification./ Think if this has to do with a change in your study patterns or style.*"
2	difficulty	trend	"You found the difficulty of the lab exercises to *increase/decrease/remain at the same level* over the weeks. *Make sure that you have understood the taught material and don't hesitate to ask for clarification./ Think if this has to do with a change in your study patterns or style.*"
Continued on next page			

Table A.1 – continued from previous page

ID	Factor	Type of template	Surface text
3	difficulty	average2	"You faced *many/a few/no* difficulties when solving the exercises. *Make sure that you have understood the taught material and don't hesitate to ask for clarification./ Think if this has to do with a change in your study patterns or style.*"
4	difficulty	other	"You found the level of difficulty of the lab exercises *very high/too low/of average* difficulty. *Make sure that you have understood the taught material and don't hesitate to ask for clarification./ Think if this has to do with a change in your study patterns or style.*"
5	hs	average	"You spent *dtime* hours studying the lecture material on average.*Keep up the good work! / Have a think about what is preventing you from studying."*
6	hs	other	"You *dedicated/did not dedicated* much time studying the lecture material *Keep up the good work! / Have a think about what is preventing you from studying."*
7	hs	trend	"You dedicated *more/less* time studying the lecture material in the beginning of the semester compared to the end of the semester. *Keep up the good work! / Have a think about what is preventing you from studying.*"

Continued on next page

Table A.1 – continued from previous page

ID	Factor	Type of template	Surface text
8	und	average	"You feel that you understood the material *completely/well enough/poorly. You should study harder to improve your comprehension of the material./ Keep up the good work! / Try going over the teaching material again.*"
9	und	other	"Your comprehension of the material is *good/enough/average. You should study harder to improve your comprehension of the material./ Keep up the good work! / Try going over the teaching material again.*"
10	und	trend	"You seem to find the material *easier/ harder* to understand compared to the beginning of the semester. *You should study harder to improve your comprehension of the material./ Keep up the good work! / Try going over the teaching material again.*"
11	deadlines	weeks	"You *haven't / have* had other deadlines *weeks. You may want to plan your studying and work ahead. / You could revise the material during the less busy weeks.*"
12	deadlines	average	"You *spent/did not spend* much time coping with other deadlines. *You may want to plan your studying and work ahead. / You could revise the material during the less busy weeks.*"

Continued on next page

Table A.1 – continued from previous page

ID	Factor	Type of template	Surface text
13	dead-lines	weeks2	"You *were/weren't* very busy during *weeksBusy. You may want to plan your studying and work ahead. / You could revise the material during the less busy weeks.*"
14	dead-lines	trend	"Your workload is *increasing/decreasing* over the semester. *You may want to plan your studying and work ahead. / You could revise the material during the less busy weeks.*"
15	hi	weeks	"You *faced some/faced some severe/did not face any* health problems at *weeksHI. You may find it useful to talk to your mentor or student welfare.*"
16	hi	average	You *faced some/faced some severe/did not face any* health problems during the semester. *You may find it useful to talk to your mentor or student welfare.*"
17	hi	trend	"You faced some health issues during the *first/second* half of the semester. / Your health condition remained stable during the semester. *You may find it useful to talk to your mentor or student welfare.*"
18	pi	weeks	"You *faced some/faced some severe/did not face any* personal issues at *weeksHI. You may find it useful to talk to your mentor or student welfare.*"
19	pi	average	"You *faced some/faced some severe/did not face any* personal issues during the semester. *You may find it useful to talk to your mentor or student welfare.*"

Continued on next page

Table A.1 – continued from previous page

ID	Factor	Type of template	Surface text
20	pi	trend	"You faced *some severe / some* personal issues at *weeksHI*, but then your condition was improved. *You may find it useful to talk to your mentor or student welfare.*"
21	la	average	"You attended *almost all / just a few* lectures during the semester. *Have a think about how to use time in lectures to improve your understanding of the material. / You should make the most of these hours, so try not to miss classes. /Make sure you have covered the material of the classes you missed.*"
22	la	weeks	"You *did not attend* lectures on *weeksNO* , but you attended *weeksYes* during the other weeks. *Have a think about how to use time in lectures to improve your understanding of the material. / You should make the most of these hours, so try not to miss classes. /Make sure you have covered the material of the classes you missed.*"
23	la	trend	"Your attendance was *increasing /decreasing* over time. *Have a think about how to use time in lectures to improve your understanding of the material. / You should make the most of these hours, so try not to miss classes. /Make sure you have covered the material of the classes you missed.*"

Continued on next page

Table A.1 – continued from previous page

ID	Factor	Type of template	Surface text
24	revision	average	"You revised *all/part of/none of* the learning material. *Have a think whether revising has improved your performance. / Think about what has affected your performance and what you can do to improve.*"
25	revision	other	"Revising material during the semester will improve your performance in the lab."
26	m	average	"Your overall performance was *excellent /poor* during the semester. *Keep up the good work and maybe try some more challenging exercises. / Make sure you revise the learning.*"
27	m	trend	"Your overall performance has *improved /deteriorated* since the beginning of the semester. *Keep up the good work and maybe try some more challenging exercises. /Make sure you revise the learning.*"
28	m	weeks	"You did well at *weeksWell*, but not at *weeksNo*". *Have a think about how you were working well and try to apply it to the other labs.*"

Appendix B

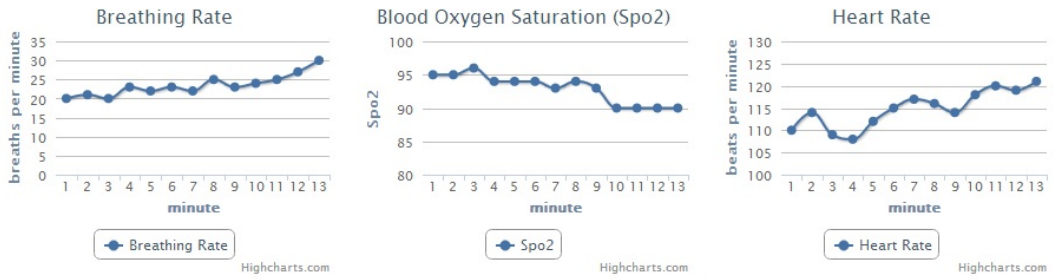
Health Informatics domain: The MIME scenarios

This appendix presents the four first aid scenarios. Each figure presents:

- On top: a textual description of the first aid scenario,
- In the middle: three graphical representations of the corresponding time-series physiological data, and
- On the bottom: the 18 available templates.

Scenario 1:

A female aged 30 years has been rescued from a burning building by Fire Service personnel. She is conscious and breathing. She has no obvious burns but is suffering from smoke inhalation and is currently being treated with 100 % oxygen by fire crews. The following graphs show the measurements of her breathing rate, blood oxygen saturation and heart rate. Which one of the given textual descriptions would you use to describe the data of each graph?



- The breathing rate was 23 breaths per minute on average.
- The breathing rate increased from 20 to 30 breaths per minute.
- Resps ↑ from 20 to 30.
- The breathing rate was between 20 and 30 breaths per minute.
- Resps 20 - 30.
- Breathing rate problematic.

- Blood oxygen saturation remained 92% per minute.
- The Blood oxygen saturation dropped from 95% to 90%.
- SATS ↓ from 95% to 90%.
- The Blood oxygen saturation was between 90% and 96%.
- SpO2 90% - 96%.
- Blood oxygen saturation problematic.

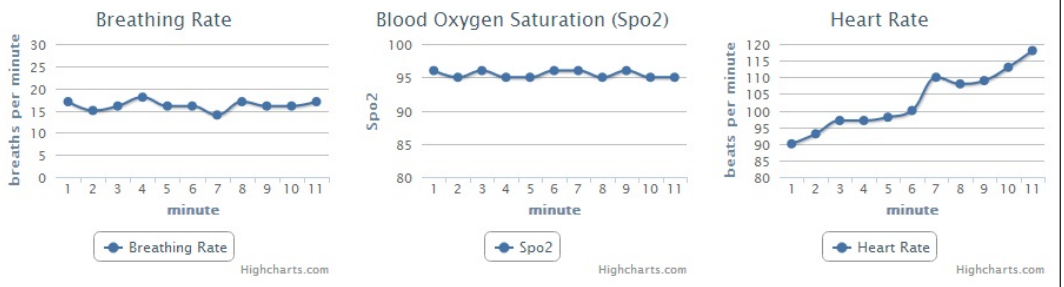
- The heart rate was 114 beats per minute on average.
- The heart rate increased from 110 to 121.
- Heart rate ↑ from 110 to 121.
- The heart rate was between 108 and 121 beats per minute.
- Heart rate 108 - 121.
- Heart rate observation problematic.

Submit

Figure B.1: The smoke inhalation scenario.

Scenario 2:

A female aged 47 years has fallen in to water whilst disembarking from a small boat onto a landing stage. Her companion saw her disappear under water and pulled her out. The patient has swallowed water, is coughing and is disoriented. The patient is now conscious and breathing, and it didn't look like she hit her head when she fell. Medical sensors were used to measure her breathing rate, blood oxygen saturation and heart rate.Which one of the given textual descriptions would you use to describe the data of each graph?



- The breathing rate was 16 breaths per minute on average.
- The breathing rate from 17 to 17 breaths per minute.
- Resps from 17 to 17.
- The breathing rate was between 14 and 18 breaths per minute.
- Resps 14 - 18.
- Breathing rate normal.

- Blood oxygen saturation remained 95% per minute.
- The Blood oxygen saturation dropped from 96% to 95%.
- SATS ↓ from 96% to 95%.
- The Blood oxygen saturation was between 95% and 96%.
- SpO2 95% - 96%.
- Blood oxygen saturation problematic.

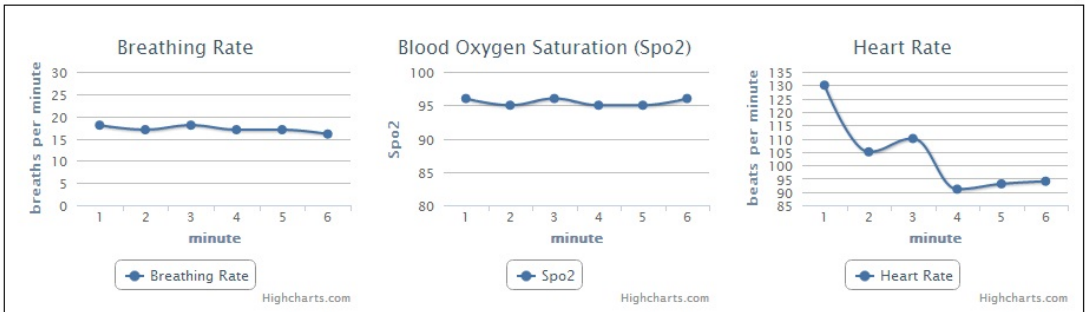
- The heart rate was 103 beats per minute on average.
- The heart rate increased from 90 to 118 beats per minute.
- Heart rate ↑ from 90 to 118.
- The heart rate was between 90 and 118 beats per minute.
- Heart rate 90 - 118.
- Heart rate observation problematic.

Submit

Figure B.2: The drowning scenario.

Scenario 3:

An elderly male (75 years) who lives on his own has stumbled down the last few steps of his stairs and fallen badly. Medical sensors were used to measure his breathing rate, blood oxygen saturation and heart rate.Which one of the given textual descriptions would you use to describe the data of each graph?



- The breathing rate was 17 breaths per minute on average.
- The breathing rate dropped from 18 to 16 breaths per minute.
- Resps ↓ from 18 to 16.
- The breathing rate was between 16 and 18 breaths per minute.
- Resps 16 - 18.
- Breathing rate normal.

- Blood oxygen saturation remained 95% per minute.
- The Blood oxygen saturation from 96% to 96%.
- SATS from 96% to 96%.
- The Blood oxygen saturation was between 95% and 96%.
- SpO2 95% - 96%.
- Blood oxygen saturation problematic.

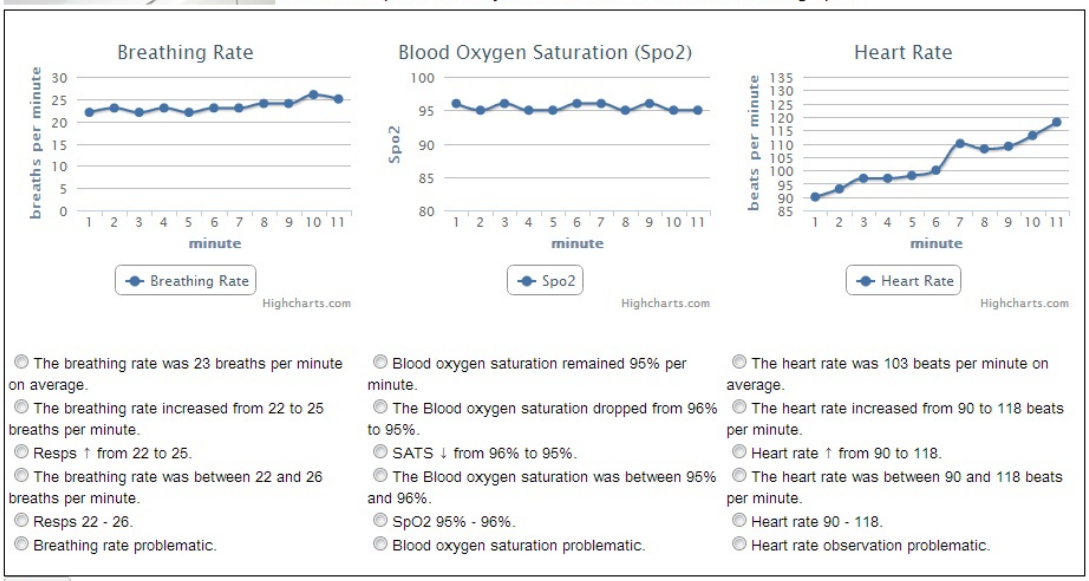
- The heart rate was 103 beats per minute on average.
- The heart rate dropped from 130 to 94 beats per minute.
- Heart rate ↓ from 130 to 94.
- The heart rate was between 91 and 130 beats per minute.
- Heart rate 91 - 130.
- Heart rate observation problematic.

Submit

Figure B.3: The fall down stairs scenario.

Scenario 4:

A male cyclist in his twenties has collided head-on with a stationary car in the main street of a rural village. His chain came loose and went through the back wheel. The patient is bleeding from their head and there appears to be some blood around his lower legs. The cyclist was traveling around 20 mph; he was not wearing a helmet. Medical sensors were used to measure his breathing rate, blood oxygen saturation and heart rate.Which one of the given textual descriptions would you use to describe the data of each graph?



- The breathing rate was 23 breaths per minute on average.
- The breathing rate increased from 22 to 25 breaths per minute.
- Resps ↑ from 22 to 25.
- The breathing rate was between 22 and 26 breaths per minute.
- Resps 22 - 26.
- Breathing rate problematic.

- Blood oxygen saturation remained 95% per minute.
- The Blood oxygen saturation dropped from 96% to 95%.
- SATS ↓ from 96% to 95%.
- The Blood oxygen saturation was between 95% and 96%.
- SpO2 95% - 96%.
- Blood oxygen saturation problematic.

- The heart rate was 103 beats per minute on average.
- The heart rate increased from 90 to 118 beats per minute.
- Heart rate ↑ from 90 to 118.
- The heart rate was between 90 and 118 beats per minute.
- Heart rate 90 - 118.
- Heart rate observation problematic.

Submit

Figure B.4: The bicycle accident scenario.

Appendix C

Reward Functions for Student Feedback

The table below describes $X = \{x_1, x_2, ..., x_n\}$. `Diff` stands for difficulty, `hs` for hours studied, `und` for understandability, `dl` for deadlines, `hi` for health issues, `pi` for personal issues, `la` for lecturers attended, `rev` for revision and `m` for marks. The fourth column contains the coefficients b_i from the Lecturer-adapted reward function resulted after analysis with multivariate regression, with adjusted $R - squared = 0.6517$ and the $p - value = 0.0001665$. The fifth column contains the coefficients w_i from the Student-adapted reward function, with adjusted $R - squared = 0.6315$ and the $p - value = 0.0002258$. The symbol N/A denotes that a coefficient cannot be estimated.

x_n	**Factor trend**	**Template type**	b_i	w_i
intercept	N/A	N/A	-81	-109
x_0	diff increase	average	-81.278	28.081
x_1	diff increase	trend	-80.532	-15.666
x_2	diff decrease	average	-77.119	-8.009

Continued on next page

Table C.1 – continued from previous page

x_n	**Factor trend**	**Template type**	b_i	w_i
x_3	diff decrease	trend	-42.483	-17.786
x_4	diff stable	average	N/A	N/A
x_5	diff stable	trend	N/A	N/A
x_6	hs increase	average	122.066	-1.388
x_7	hs increase	trend	N/A	-4.104
x_8	hs decrease	average	146.369	-21.545
x_9	hs decrease	trend	137.747	- 57.403
x_{10}	hs stable	average	155.035	-13.136
x_{11}	hs stable	trend	205.309	-9.206
x_{12}	und increase	average	-13.904	91.084
x_{13}	und increase	trend	-3.3	108.877
x_{14}	und decrease	average	-50.7	-48.928
x_{15}	und decrease	trend	-26.446	-67.335
x_{16}	und stable	average	N/A	N/A
x_{17}	und stable	trend	N/A	N/A
x_{18}	dl increase	average	57.791	-17.520
x_{19}	dl increase	weeks	59.628	-20.573
x_{20}	dl increase	trend	41.866	-18.150
x_{21}	dl decrease	average	N/A	N/A
x_{22}	dl decrease	weeks	N/A	N/A
x_{23}	dl decrease	trend	N/A	N/A
x_{24}	dl stable	average	N/A	N/A
x_{25}	dl stable	weeks	N/A	N/A
x_{26}	dl stable	trend	N/A	N/A

Continued on next page

Table C.1 – continued from previous page

x_n	**Factor trend**	**Template type**	b_i	w_i
x_{27}	hi increase	weeks	-18.84	-35.094
x_{28}	hi increase	trend	-136.933	-56.389
x_{29}	hi decrease	weeks	5.491	-17.605
x_{30}	hi decrease	trend	N/A	-126.928
x_{31}	hi stable	weeks	40.0	-25.157
x_{32}	hi stable	trend	-36.119	-46.090
x_{33}	pi increase	weeks	19.756	-66.108
x_{34}	pi increase	trend	233.037	47.557
x_{35}	pi decrease	weeks	34.876	-67.000
x_{36}	pi decrease	trend	-1.476	-42.873
x_{37}	pi stable	weeks	41.784	-99.506
x_{38}	pi stable	trend	-10.487	37.168
x_{39}	la increase	average	22.973	-85.128
x_{40}	la increase	weeks	na	N/A
x_{41}	la increase	trend	16.623	-83.757
x_{42}	la decrease	average	15.676	-12.487
x_{43}	la decrease	weeks	1.631	-10.497
x_{44}	la decrease	trend	31.84	12.831
x_{45}	la stable	average	28.614	-50.702
x_{46}	la stable	weeks	20.142	-25.581
x_{47}	la stable	trend	41.209	-19.150
x_{48}	rev increase	average	-62.734	N/A
x_{49}	rev increase	other	-88.143	-99.111
x_{50}	rev decrease	average	-88.164	-75.815

Continued on next page

Table C.1 – continued from previous page

x_n	**Factor trend**	**Template type**	b_i	w_i
x_{51}	rev decrease	other	-81.814	-42.351
x_{52}	rev stable	average	-49.524	-89.238
x_{53}	rev stable	other	-88.603	-51.791
x_{54}	m increase	average	66.412	-39.282
x_{55}	m increase	weeks	85.695	N/A
x_{56}	m increase	trend	68.836	-76.969
x_{57}	m decrease	average	92.586	-68.297
x_{58}	m decrease	weeks	88.973	N/A
x_{59}	m decrease	trend	77.29	-86.127
x_{60}	m stable	average	85.224	-150.965
x_{61}	m stable	weeks	-46.444	N/A
x_{62}	m stable	trend	N/A	-66.783
x_{63}	diff increase	non mentioned	-56.188	118.901
x_{64}	diff decrease	non mentioned	-48.345	42.200
x_{65}	diff stable	non mentioned	N/A	N/A
x_{66}	hs increase	non mentioned	131.120	33.260
x_{67}	hs decrease	non mentioned	130.966	7.678
x_{68}	hs stable	non mentioned	117.423	N/A
x_{69}	und increase	non mentioned	-23.854	20.867
x_{70}	und decrease	non mentioned	-22.151	-22.013
x_{71}	und stable	non mentioned	N/A	N/A
x_{72}	dl increase	non mentioned	39.123	26.551
x_{73}	dl decrease	non mentioned	N/A	N/A
x_{74}	dl stable	non mentioned	N/A	N/A

Continued on next page

Table C.1 – continued from previous page

x_n	**Factor trend**	**Template type**	b_i	w_i
x_{75}	hi increase	non mentioned	2.803	36.989
x_{76}	hi decrease	non mentioned	12.142	-11.111
x_{77}	hi stable	non mentioned	na	N/A
x_{78}	pi increase	non mentioned	10.796	46.405
x_{79}	pi decrease	non mentioned	10.291	20.434
x_{80}	pi stable	non mentioned	N/A	32.615
x_{81}	la increase	non mentioned	22.984	14.963
x_{82}	la decrease	non mentioned	8.663	82.468
x_{83}	la stable	non mentioned	N/A	68.410
x_{84}	rev increase	non mentioned	-101.075	-33.129
x_{85}	rev decrease	non mentioned	-79.26	-14.784
x_{86}	rev stable	non mentioned	-71.341	N/A
x_{87}	m increase	non mentioned	-40.599	-7.766
x_{88}	m decrease	non mentioned	24.744	-6.734
x_{89}	m stable	non mentioned	na	N/A
length	N/A	N/A	-3.376	51.859

Appendix D

Rule-based System for Feedback Generation

This appendix presents the pseudo-code for the rule-based system used for Feedback Generation as described in Chapter 4.

```
Input: Templates t, Student s, time-series data trends (0, 1, 2)
Output: feedback

  //marks
  IF (s.marks() = 0 & s.MarksAverage() > 0)
        feedback += tl.getTemplate(26)
  ELSE IF (s.marks() = 1)
        feedback += tl.getTemplate(27)
  ELSE
        feedback += tl.getTemplate(25)

  //lectures attended
  IF(s.lecturesAttended() = 0 )
        feedback += tl.getTemplate(21)
  ELSE IF (s.lecturesAttended() =1)
        feedback += tl.getTemplate(22)
  ELSE IF (s.lecturesAttended() = 2 & s.LectAverage() = 3)
        feedback += tl.getTemplate(20)

  //difficulty
  IF (s.difficulty() = 0 )
        feedback += tl.getTemplate(3)
  ELSE IF (s.difficulty() = 1)
        feedback += tl.getTemplate(3)
  ELSE
        feedback += tl.getTemplate(3)
```

```
//hours studied
IF(s.hoursStudied() = 0 )
      feedback += tl.getTemplate(4)
ELSE IF (s.hoursStudied() = 1)
      feedback += tl.getTemplate(6)
ELSE IF (s.hoursStudied() = 2)
      feedback += tl.getTemplate(5)

//understandability
IF (s.understandability() = 0 & s.UnderstandabilityAverage() > 4)
      feedback += tl.getTemplate(8)
ELSE IF (s.understandability() = 1)
      feedback += tl.getTemplate(7)
ELSE IF (s.understandability() = 2 & s.UnderstandabilityAverage() >= 4)
      feedback += tl.getTemplate(8)
ELSE IF (s.understandability() = 2 & s.UnderstandabilityAverage() < 4)
      feedback += tl.getTemplate(7)

//deadlines
IF (s.deadlines() = 0 )
      feedback += tl.getTemplate(14)
ELSE IF (s.deadlines() = 1)
      feedback += tl.getTemplate(11)
ELSE IF (s.deadlines() = 2 & s.DeadlinesAverage() >= 2)
      feedback += tl.getTemplate(10)

//health issues
IF(s.healthIssues() = 0)
      feedback += tl.getTemplate(16)
ELSE IF (s.healthIssues() = 1)
      feedback += tl.getTemplate(16)
ELSE IF (s.HealthAverage() == 2 & s.HealthAverage() > 2)
      feedback += tl.getTemplate(15)

//personal issues
IF (s.personalIssues()  =  0 )
      feedback += tl.getTemplate(19)
ELSE IF (s.personalIssues() = 1)
      feedback += tl.getTemplate(19)
ELSE IF (s.personalIssues() = 2 & s.PersonalAverage() > 2)
      feedback += tl.getTemplate(18)

//Revision
IF (s.revision() =  0 )
      feedback += tl.getTemplate(23)
ELSE IF (s.revision() = 1)
      feedback += tl.getTemplate(23)
ELSE IF (s.revision() = 2 & s.MarksAverage() < 3)
      feedback += tl.getTemplate(24)
```

Appendix E

Examples of Decision Trees

Algorithm 6: Rules derived from J48 decision trees for template 26 (marks - describing the trend).

```
if lectures_attended: increasing OR lectures_attended: other then
    Generate template_26;
else if lectures_attended: decreasing AND marks: increasing then
    Generate template_26;
```

Algorithm 7: Rules derived from J48 decision trees for template 26 including history, i.e. previous decisions.

```
if template_9 = 0 then
    if template_1 = 0 then
        if revision: increasing AND template_6 = 0 AND (hours_studied: decreasing
        OR other) then
            Generate template_26 ;
    else
        Generate template_26;
    end
```

Bibliography

Ames, C. (1992). Classrooms: Goals, Structures, and Student Motivation. *Journal of Educational Psychology*, 84(3):261–71.

Angeli, G., Liang, P., and Klein, D. (2010). A simple domain-independent probabilistic approach to generation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 502–512.

Banaee, H., Ahmed, M. U., and Loutfi, A. (2013). Towards NLG for Physiological Data Monitoring with Body Area Networks. In *14th European Workshop on Natural Language Generation (ENLG)*, pages 193–197.

Barzilay, R. and Lapata, M. (2005). Collective content selection for concept-to-text generation. In *Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT - EMNLP)*, pages 331–338.

Barzilay, R. and Lee, L. (2004). Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT - NAACL)*, pages 113–120.

Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press.

Belz, A. and Hastie, H. (2014). *Towards Comparative Evaluation and Shared Tasks for NLG in Interactive Systems*. Cambridge University Press.

Belz, A. and Kow, E. (2010). Extracting parallel fragments from comparable corpora for data-to-text generation. In *6th International Natural Language Generation Conference (INLG)*, pages 167–171.

Belz, A. and Reiter, E. (2006). Comparing Automatic and Human Evaluation of NLG Systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics (ACL)*, pages 313–320.

Black, R., Reddington, J., Reiter, E., Tintarev, N., and Waller, A. (2010). Using NLG and Sensors to Support Personal Narrative for Children with Complex Communication Needs. In *NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies*, pages 1–9.

Bouayad-Agha, N., Casamayor, G., Wanner, L., and Mellish, C. (2012). Content Selection from Semantic Web Data. In *7th International Natural Language Generation Conference (INLG)*, pages 146–149.

Boutilier, C. (2013). Computational Decision Suppot: Regret-based Models for Optimization and Preference Elicitation. Technical report, University of Toronto.

Boyd, S. (1998). TREND: A System for Generating Intelligent Descriptions of Time-Series Data. In *IEEE International Conference on Intelligent Processing Systems*.

Branke, J., Deb, K., Dierolf, H., and Osswald, M. (2004). Finding Knees in Multi-objective Optimization. *Parallel Problem Solving from Nature - Lecture Notes in Computer Science*, 3242:722 – 731.

Chi, M. T., Siler, S. A., Jeong, H., Yamauchi, T., and Hausmann, R. G. (2001). Learning from Human Tutoring. *Journal of Cognitive Science*, 25(4):471–533.

Craig, S. D., Graesser, A. C., Sullins, J., and Gholson, B. (2004). Affect and learning: an exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media*, 29:241–250.

Darwin, C. (1909). *The foundations of the Origins of Species*. Cambridge University Press.

Deb, K. (2001). *Multi-objective Optimization using Evolutionary Algorithms*. Wiley.

Demberg, V. and Moore, J. (2006). Information Presentation in Spoken Dialogue Systems. In *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Demir, S., Carberry, S., and McCoy, K. (2011). Summarizing Information Graphics Textually. *Computational Linguistics*, 38(3):527 – 574.

Dethlefs, N. and Cuayahuitl, H. (2011). Combining hierarchical reinforcement learning and bayesian networks for natural language generation in situated dialogue. In *13th European Workshop on Natural Language Generation (ENLG)*, pages 110–120.

Dethlefs, N., Cuayahuitl, H., Hastie, H., Rieser, V., and Lemon, O. (2014). Cluster-based Prediction of User Ratings for Stylistic Surface Realisation. In *14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 702–711.

DiMarco, C., Bray, P., Covvey, D., Cowan, D., DiCiccio, V., Lipa, J., and Yang, C. (2008). Authoring and Generation of Individualised Patient Education Materials. *Information Technology in Healthcare*, 6(1):63-71.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *2nd International Conference on Human Language Technology Research (HLT)*, pages 138–145.

Duboue, P. and McKeown, K. (2003). Statistical acquisition of Content Selection Rules for Natural Language Generation. In *Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT - EMNLP)*.

Duboue, P. and McKeown, K. R. (2002). Content Planner Construction via Evolutionary Algorithms and a Coprus-based Fitness Function. In *2nd International Natural Language Generation Conference (INLG)*.

Foster, M. E. (2008). Automated metrics that agree with human judgements on generated output for an embodied conversational agent. In *5th International Natural Language Generation Conference (INLG)*.

Foster, M. E. and Oberlander, J. (2006). Data-driven generation of emphatic facial displays. In *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 353–360.

Foster, M. E. and Oberlander, J. (2007). Corpus-based generation of head and eyebrow motion for an embodied conversational agent. *Language Resources and Evaluation*, 41(3):305 – 323.

Fox, B. (1993). *The Human Tutorial Dialogue Project: Issues in the Design of Instructional Systems*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Gatt, A., Portet, F., Reiter, E., Hunter, J., Mahamood, S., Moncur, W., and Sripada, S. (2009). From Data to Text in the Neonatal Intensive Care Unit: Using NLG Technology for Decision Support and Information Management. *AI Communications*, 22: 153-186.

Gkatzia, D. (2013). "Keep up the good work!" Generating Feedback for Students using Reinforcement Learning. In *SICSA PhD Conference*.

Gkatzia, D. and Hastie, H. (2012). Dynamic user modelling for personalized report generation of time-series data. In *Symposium on Influencing People with Information (SIPI)*.

Gkatzia, D. and Hastie, H. (2015). An Ensemble Method for Content Selection for Data-to-text Generation. In *1st International Workshop on Data-to-text Generation*.

Gkatzia, D., Hastie, H., Janarthatanam, S., and Lemon, O. (2013). Generating student feedback from time-series data using Reinforcement Learning. In *In Proceedings of the 14th European Workshop on Natural Language Generation (ENLG)*, pages 115–124.

Gkatzia, D., Hastie, H., and Lemon, O. (2014a). Comparing Multi-label classification with Reinforcement Learning for Summarisation of Time-series data. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1231 – 1240.

Gkatzia, D., Hastie, H., and Lemon, O. (2014b). Finding Middle Ground? Multi-objective Natural Language Generation from time-series data. In *14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 210–214.

Gkatzia, D., Hastie, H., and Lemon, O. (2014c). Multi-adaptive Natural Language Generation using Principal Component Regression. In *8th International Natural Language Generation Conference (INLG)*, pages 138–142.

Gkatzia, D., Rieser, V., McSporran, A., McGowan, A., Mort, A., and Dewar, M. (2014d). Generating Verbal Descriptions from Medical Sensor Data: A Corpus Study on User Preferences. In *BCS Health Informatics Scotland (HIS)*.

Grice, P. (1975). Logic and conversation. *In Syntax and Semantics*, 3.

Hallett, C., Power, R., and Scott, D. (2006). Summarisation and visualisation of e-health data repositories. In *UK E-Science All-Hands Meeting*.

Han, X., Sripada, S., Macleod, K. C., and Ioris, A. A. R. (2014). Latent User Models for Online River Information Tailoring. In *8th International Natural Language Generation Conference (INLG)*, pages 133–137.

Handl, J. and Knowles, J. (2008). *Modes of Problem Solving with Multiobjective Optimization: Implications for Interpreting the Pareto Set and for Decision Making*, pages 131 – 151. Springer Natural Computing Series, Springer-Verlag.

HEA (2009). Providing individual written feedback on formative and summative assessments.

Hunter, J., Freer, Y., Gatt, A., Sripada, Y., Sykes, C., and Westwater, D. (2011). BT-Nurse: Computer Generation of Natural Language Shift Summaries from Complex Heterogeneous Medical Data. *American Medical Informatics Association*, 18(5):621-624.

Hutchins, W. J. and Somers, H. L. (1992). *An introduction to Machine Translation*. Academic Press.

Janarthanam, S. (2011). *Learning user modelling strategies for adaptive referring expression generation in spoke dialogue systems*. PhD thesis, University of Edinburgh.

Janarthanam, S. and Lemon, O. (2010). Adaptive Referring Expression Generation in Spoken Dialogue Systems: Evaluation with Real Users. In *11th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 124–131.

Johnson, N. and Lane, D. (2011). Narrative monologue as a first step towards advanced mission debrief for AUV operator situational awareness. In *15th International Conference on Advanced Robotics*.

Jolliffe, I. T. (1982). A note of the Use of Principal Components in Regression. *Royal Statistical Society, Series C*, 31(3):300 – 303.

Knight, K. and Hatzivassiloglou, V. (1995). Two-Level, Many-Paths Generation. In *Conference of the Association for Computational Linguistics (ACL)*, pages 252–260.

Kondadadi, R., Howald, B., and Schilder, F. (2013). A Statistical NLG Framework for Aggregated Planning and Realization. In *51st Annual Meeting of the Association for Computational Linguistics*, pages 1406–1415.

Konstas, I. and Lapata, M. (2012). Unsupervised concept-to-text generation with hypergraphs. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 752–761.

Kukich, K. (1983). Design of a knowledge-based report generator. In *21st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 145–150.

Lampouras, G. and Androutsopoulos, I. (2013). Using Integer Linear Programming in Concept-to-Text Generation to Produce More Compact Texts. In *51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 561–566.

Langkilde, I. and Knight, K. (1998). Generation that exploits Coprus-based Statistical Knowledge. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (ACL)*, pages 704–710.

Law, A. S., Freer, Y., Hunter, J., Logie, R. H., McIntosh, N., and Quinn, J. (2005). A Comparison of Graphical and Textual Presentations of Time Series Data to Support Medical Decision Making in the Neonatal Intensive Care Unit. *Journal of Clinical Monitoring and Computing*, pages 19: 183–194.

Lemon, O. (2011). Learning what to say and how to say it: joint optimization of spoken dialogue management and Natural Language Generation. *Computer Speech and Language*, 25(2):210–221.

Liang, P., Jordan, M. I., and Klein, D. (2009). Learning semantic correspondences with less supervision. In *Joint Conference of the 47th Annual Meeting of the Association of Computational Linguistics (ACL) and the 4th International Joint Conference on Natural Language Processing (IJNLP)*, pages 91–99.

Lin, C.-Y. (2004). ROUGE: a package for automatic evaluation of summaries. In *Workshop on Text Summarization (WAS)*, pages 25–26.

Loomes, G. and Sugden, R. (1982). Regret Theory: An Alternative Theory of Rational Choice Under Uncertainty. *The Economic Journal*, 92(368):805–824.

Madjarov, G., Kocev, D., Gjorgjevikj, D., and Dzeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9):3084–3104.

Mahamood, S., Bradshaw, W., and Reiter, E. (2014). Generating Annotated Graphs using the NLG Pipeline Architecture. In *8th International Natural Language Generation Conference (INLG)*.

Mahamood, S. and Reiter, E. (2011). Generating affective natural language for parents of neonatal infants. In *13th European Workshop on Natural Language Generation (ENLG)*, pages 12–21.

Mairesse, F. and Walker, M. (2007). PERSONAGE: Personality generation for dialogue. In *45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 496–503.

Mann, W. and Thompson, S. (1988). Rhetorical Structure Theory: Toward a functional theory of text generation. *Text*, 8(3):243 – 281.

Markov, A. (1954). *Theory of Algorithms*. Imprint Moscow, Academy of Sciences of the USSR.

McKeown, K. (1985). Discourse strategies for generating natural-language text. *Computer Speech and Language*, 27:1 – 42.

Mellish, C., Knott, A., Oberlander, J., and O'Donnell, M. (1998). Experiments using stochastic search for text planning. In *International Conference on Natural Language Generation (INLG)*, pages 98–107.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Moore, J., Porayska-Pomsta, K., Varges, S., and Zinn, C. (2004). Generating Tutorial Feedback with Affect. In *17th International Florida Artificial Intelligence Research Society Conference, AAAI Press*.

O'donnell, M., Mellish, C., Oberlander, J., and Knott, A. (2001). ILEX: an architecture for a dynamic hypertext generation system. *Natural Language Engineering*, 7(3):225 – 250.

Olson, D. and Delen, D. (2008). *Advanced Data Mining Techniques*. Springer.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Peddington, J. and Tintarev, N. (2011). Automatically generating stories from sensor data. In *6th International Conference on Intelligent user Interfaces (IUI)*, pages 407–410.

Person, N. K., Kreuz, R. J., Zwaan, R. A., and Graesser, A. C. (1995). Pragmatics and Pedagogy: Conversational Rules and Politeness Strategies May Inhibit Effective Tutoring. *Journal of Cognition and Instruction*, 13(2):161–188.

Petre, M. (1995). Why looking isn't always seeing: Readership skills and Graphical Programming. *Transactions of the ACM*, 38(6):33–44.

Porayska-Pomsta, K. and Mellish, C. (2013). Modelling human tutors' feedback to inform natural language interfaces for learning. *International Journal of Human-Computer Studies*, 71(6):703–724.

Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., and Sykes, C. (2007). Automatic generation of textual summaries from neonatal intensive care data. In *11th Conference on Artificial Intelligence in Medicine (AIME)*, volume 789-816.

Rambow, O., Rogati, M., and Walker, M. (2001). Evaluating a Trainable Sentence Planner for a Spoken Dialogue System. In *39th Meeting of the Association for Computational Linguistics (ACL)*, pages 426–433.

Reiter, E. (2007). An Architecture for Data-to-Text Systems. In *11th European Workshop on Natural Language Generation (ENLG)*, pages 97–104.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation systems*. Cambridge University Press.

Reiter, E., Robertson, R., and Osman, L. (1999). Types of knowledge required to personalise smoking cessation letters. In *Artificial Intelligence in Medicine: Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, pages 389–399.

Reiter, E., Spirada, S. G., and Robertson, R. (2003). Acquiring correct knowledge for natural language generation. *Artificial Intelligence Research (JAIR)*, 18:491–516.

Reiter, E. and Sripada, S. (2002). Should Corpora Texts Be Gold Standards for NLG? In *2nd International Natural Language Generation Conference (INLG)*, pages 97–104.

Rieser, V. and Lemon, O. (2011). *Reinforcement Learning for Adaptive Dialogue Systems: A Data-driven Methodology for Dialogue Management and Natural Language Generation*. Theory and Applications of Natural Language Processing, Springer.

Rieser, V., Lemon, O., and Liu, X. (2010). Optimising Information Presentation for Spoken Dialogue Systems. In *48th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Schaffer, D. J. (1985). Multiple Objective Optimizatio Using Nondominated Sorting in Genetic Algorithms. *Ecolutionary Computation*, 2(3):221–248.

Schneider, A., Vaudry, P.-L., Mort, A., Mellish, C., Reiter, E., and Wilson, P. (2013). MIME - NLG in Pre-hospital Care. In *14th European Workshop on Natural Language Generation (ENLG)*, pages 152–156.

Skinner, B. F. (1938). *The behavior of Organisms: An Experimental Analysis*. B.F. Skinner Foundation.

Sowdaboina, P. K. V., Chakraborti, S., and Sripada, S. (2014). Learning to Summarize Time Series Data. *Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science*, 8403:515 – 528.

Sripada, S. and Gao, G. (2007). Summarizing dive computer data: A case study in integrating textual and graphical presentations of numerical data. In *Workshop on Multimodal Output Generation (MOG)*.

Sripada, S., Reiter, E., Davy, I., and Nilssen, K. (2004). Lessons from Deploying NLG Technology for Marine Weather Forecast Text Generation. In *PAIS session of ECAI-2004:760-764*.

Sripada, S., Reiter, E., Hunter, J., and Yu, J. (2003). Generating English Summaries of Time Series Data using the Gricean Maxims. In *9th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 187–196.

Sripada, S. G., Reiter, E., and Hawizy, L. (2005). Evaluation of an NLG system using post-edit data. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1700–1701.

Sripada, S. G., Reiter, E., Hunter, J., and Yu, J. (2001). A two-stage model for content determination. In *8th European workshop on Natural Language Generation (ENLG)*.

Stedt, A. J. (2011). Computational Approaches to the Production of Referring Expressions: Dialog Changes (Almost) Everything. In *PRE-CogSci Workshop*.

Stent, A., Prasad, R., and Walker, M. (2004). Trainable Sentence Planning for Complex Information Presentation in Spoken Dialog Systems. In *Proceedings of Association for Computational Linguistics (ACL)*, pages 79–86.

Stock, O., Zancanaro, M., Busetta, P., Callaway, C., Krger, A., Kruppa, M., Kuflik, T., Not, E., and Rocchi, C. (2007). Adaptive, intelligent presentation of information for the museum visitor in PEACH. *User Modeling and User-Adapted Interaction*, 17(3):257–304.

Sullivan, G. M. and Feinn, R. (2012). Using Effect Size - or Why the P Value Is Not Enough. *Graduate Medical Education*, 4(3):279–282.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning*. MIT Press.

Thomas, K., Sripada, S., and Noordzij, M. L. (2010). Atlas.txt: Exploring Linguistic Grounding Techniques for Communicating Spatial Information to Blind Users. In *Universal Access in the Information Society. DOI: 10.1007/s10209-010-0217-5*.

Thompson, C. A., Goker, M. H., and Langley, P. (2004). A personalised system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21(1):393 – 428.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 3(3):1–13.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). Random k-Labelsets for Multi-Label Classification. *IEEE Transactions on Knowledge and Data Engineering*, 99(1):1079–1089.

Tsoumakas, G., Soyrotmitros-Xioufis, E., Vilcek, J., and Vlahavas, I. (2011). MULAN: A Java Library for Multi-Label Learning. *Journal of Machine Learning Research*, 12(1):2411 – 2414.

Turner, R., Sripada, S., Reiter, E., and Davy, I. (2008). Using Spatial Reference Frames to Generate Grounded Textual Summaries of Georeferenced Data. In *5th International Natural Language Generation Conference (INLG)*, pages 16–24.

van den Meulen, M., Logie, R., Freer, Y., Sykes, C., McIntosh, N., and Hunter, J. (2010). When a graph is poorer than 100 words: A comparison of computerised natural language generation, human generated descriptions and graphical displays in neonatal intensive care. *Applied Cognitive Psychology*, 24:77 – 89.

Walker, M., Kamm, C., and Litman, D. (2000). Towards developing general models of usability with paradise. *Natural Language Engineering*, 6(3):363 – 377.

Walker, M., Stent, A., Mairesse, F., and Prasad, R. (2007). Individual and Domain Adaptation in Sentence Planning for Dialogue. *Artificial Intelligence Research (JAIR)*, 30:413 – 456.

Walker, M. A., Rambow, O. C., and Rogati, M. (2002). Training a sentence planner for spoken dialogue using boosting. *Computer Speech and Language*, 16:409 – 433.

Williams, S. and Reiter, E. (2008). SkillSum: basic skills screening with personalised, computer-generated feedback. In *11th International Conference on Interactive Computer Aided Learning*, pages 1–8.

Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers.

Yu, J., Reiter, E., Hunter, J., and Mellish, C. (2007). Choosing the content of textual summaries of large time-series data sets. *Journal Natural Language Engineering*, 13(1):25–49.

Zhang, M.-L. and Zhou, Z.-H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038 – 2048.

Zukerman, I. and Litman, D. (2001). Natural language processing and user modeling: Synergies and limitations. *In User Modeling and User-Adapted Interaction*, 11(1-2):129 – 158.